...feel the spirit of a new generation

RESI-MEDIC

Grafische Programmierung mit RESI'S GRAPH

Benutzerhandbuch Teil II

CE

BESTELL NR: RESI-GRAPH-D-20

DI Heinz-Christian SIGL:

RESI'S GRAPH - Benutzerhandbuch

Bestell Nr: RESI-GRAPH-D-20

Version: 2.00

Text, Abbildungen und Programme wurden mit größter Sorgfalt erarbeitet. Die Firma RESI, Übersetzer und Autoren können jedoch für eventuell verbliebene fehlerhafte Angaben und deren Folgen weder eine juristische Verantwortung noch irgendeine Haftung übernehmen.

Die vorliegende Publikation ist urheberrechtlich geschützt. Alle Rechte vorbehalten. Kein Teil dieses Buches darf ohne vorherige schriftliche Genehmigung der Firma RESI in irgendeiner Form durch Fotokopie, Mikrofilm oder andere Verfahren reproduziert oder in eine für Maschinen, insbesondere Datenverarbeitungsanlagen, verwendbare Sprache übertragen werden. Auch die Rechte der Wiedergabe durch Vortrag, Funk und Fernsehen sind vorbehalten.

Diese Dokumentation und die dazugehörige Software sind urheberrechtlich von der Firma RESI geschützt.

© Copyright 2002-2009 by DI Heinz-Christian SIGL

Vorsicht! Sie hantieren mit gefährlicher elektrischer Spannung!

- Schalten Sie das Gerät spannungsfrei
- Sichern Sie das Gerät gegen Wiedereinschalten
- Stellen Sie die Spannungsfreiheit des Geräts und seiner Umgebung fest
- Die für das Gerät spezifizierten Einbau und Installationshinweise sind zu beachten
- Nur entsprechend ausgebildetes und qualifiziertes Personal darf Eingriffe an diesem Gerät vornehmen
- Das Gerät ist so einzubauen, dass keine unbeabsichtigte Betätigung möglich ist.
- Alle Signal- und Anschlussleitungen sind so zu verlegen, dass induktive und kapazitive Störungen sowie Einstreuungen die Funktionen des Geräts nicht beeinflussen
- Schwankungen und Abweichungen der Netzspannung vom Nennwert dürfen die in den technischen Daten angegebenen Toleranzgrenzen und Vorgaben nicht überschreiten. Bei Nichteinhaltung kann es zu Funktionsbeeinträchtigungen und –störungen kommen
- Nach Spannungseinbrüchen oder –ausfällen darf es zu keiner Fehlfunktion in der Software des Gerätes kommen, sodaß Fehlfunktionen oder undefinierte Zustände der Anlage auftreten
- Not-Aus-Einrichtungen nach EN60204/IEC204(VDE0113) müssen in allen Betriebsarten der Automatisierungseinrichtungen wirksam bleiben. Entriegeln der Not-Aus-Einrichtung darf keinen unkontrollierten oder undefinierten Wiederanlauf bewirken.
- Bei Leitungs- oder Aderbrüchen der Signalleitungen muss softwareseitig sichergestellt werden, dass es zu keinen Fehlfunktionen kommt.

Inhaltsverzeichnis

VORSICHT! SIE HANTIEREN MIT GEFÄHRLICHER ELEKTRISCHER SPANNUNG! ... 3
INHALTSVERZEICHNIS ... 4
ZUSTÄNDE ... 12
Zustand: Zustand einfügen ... 12
Analoge Zustände ... 13
Binäre Zustände ... 14
Beispiel: Zustandsmaschine ... 15
SYSTEM ... 17
System: System Bit-Speicher ... 17
System: WENN Eingang gleich Eins SETZE Bit-Speicher ... 18
System: WENN Eingang gleich Eins LÖSCHE Bit-Speicher ... 18
System: WENN Eingang gleich Eins INVERTIERE Bit-Sp. ... 19
System: System Analog-Speicher ... 19
System: System Text-Speicher ... 20
KOMMENTARE + SYMBOLISCHE GRUPPEN ... 21
Kommentare einfügen ... 21
Symbolische Gruppe erstellen ... 21
I/O ... 23
I/O: Digitaleingäne ... 23
I/O: Digitalausgänge ... 24
I/O: Analogeingänge ... 26
I/O: Analogausgänge ... 27
I/O: Potentiometer ... 29
I/O: Texteingänge ... 30
I/O: Textausgänge ... 31
GRUPPEN ... 32

Gruppen exportieren 32
Gruppen importieren 33
Gruppen löschen 33
Regler einfügen 34
OBJEKTE 35
Objekte:Zeitrelais 35
Objekte:Zeitrrelais: Einschaltverzögerung 37
Objekte:Zeitrelais: Ausschaltverzögerung 37
Objekte:Zeitrelais: Ein/Ausschaltverzögerung 38
Objekte:Zeitrelais: Einschaltwischend 38
Objekte:Zeitrelais: Ausschaltwischend 38
Objekte:Zeitrelais: Ein/Ausschaltwischend 39
Objekte:Zeitrelais: Blinken ein beginnend 39
Objekte:Zeitrelais: Blinken aus beginnend 39
Objekte: Zeitrelais: Delay 40
ECHTZEITUHR 41
Objekte:Uhr:Exakte Uhrzeit 41
Objekte:Uhr:Uhrzeitbereich 42
Objekte:Uhr:Exaktes Datum 42
Objekte:Uhr:Datumsbereich 42
Objekte:Uhr:Exakter Zeitpunkt 43
Objekte:Uhr:Zeitbereich 43
Objekte:Uhr:Exakter Wochentag 44
Objekte:Uhr:Wochentagsbereich 44
Objekte:Uhr:Exakte Woche 44
Objekte:Uhr:Wochenbereich 45
Objekte:Uhr:Analog:Uhrzeit 45
Objekte:Uhr:Analog:Datum 46
Objekte:Uhr:Analog:Wochentag 46
Objekte:Uhr:Analog:Woche 46

Objekte:Uhr:Text:Uhrzeit 47
Objekte:Uhr:Text:Datum 47
Objekte:Uhr:Text:Datum+Uhrzeit 47
Objekte:Uhr:Text:Wochentag 48
Objekte:Uhr:Text:Woche 48
Objekte:Uhr:Zeitprogramm 48
CAN OBJEKTE 49
Objekte:CAN Meldung ohne Daten empfangen 49
Objekte:CAN Meldung mit Analogwert empfangen 50
Objekte:CAN Meldung mit Text empfangen 50
Objekte:CAN Meldung mit vollen Daten empfangen 51
Objekte:CAN Meldung ohne Daten senden 51
Objekte:CAN Meldung mit Analogwert senden 51
Objekte:CAN Meldung mit Textwert senden 52
Objekte:CAN Meldung vollen Daten senden 52
SIO FUNKTIONEN 54
Objekte:SIO:Sende Text 54
Objekte:SIO:Sende ein Byte 55
Objekte:SIO:Send ein Wort(2 Bytes) 55
Objekte:SIO:Sende ein Doppelwort(4 Bytes) 55
Objekte:SIO:Empfange ein Byte 56
Objekte:SIO :Empfange einen Text 56
TERMINALFUNKTIONEN 57
Objekte:Terminal:Meldung anzeigen 57
Objekte:Terminal:Meldung mit Wert anzeigen 59
Objekte:Terminal:Meldung mit Text anzeigen 60
Objekte:Terminal:Text eingeben 61
Objekte:Terminal:Analogwert eingeben 63
Objekte:Terminal:Menü anzeigen 65
Objekte:Terminal:Element auswählen 67

Objekte:Terminal:Analogwert anzeigen 69
Objekte:Terminal:Text anzeigen 70
Objekte:Terminal:Taste abfragen 71
SPEICHERKARTE 73
Objekte:Speicherkarte:Analogwert in Speicher lesen 73
Objekte:MemoryCard:Textwert in Speicher lesen 74
Objekte:MemoryCard:Analogwert schreiben 75
Objekte:MemoryCard:Textwert schreiben 75
Objekte:MemoryCard:Analogwert lesen 76
Objekte:MemoryCard:Texwert auslesen 76
Objekte:MemoryCard:Analogwert sofort schreiben 77
Objekte:MemoryCard:Analogwert sofort lesen 77
Objekte:MemoryCard:Textwert sofort schreiben 78
Objekte:MemoryCard:Textwert sofort lesen 78
POSITIONIERUNG 80
Objekte:Positionierung: Verfahrgeschwindigkeit 80
Objekte:Positionierung: Referenzgeschwindigkeit 81
Objekte:Positionierung: Referenzbewegung durchführen 82
Objekte:Positionierung: Positionierung auf Absolutposition durchführen 82
Objekte:Positionierung: Positionierung auf Relativposition durchführen 83
Objekte:Positionierung: Aktive Positionierung abbrechen 83
Objekte:Positionierung: Aktive Positionierung sofort stoppen 84
Objekte:Positionierung: Achsenpaare verwenden 84
Objekte:Positionierung: Aktuelle X-Position 85
Objekte:Positionierung: Aktuelle Y-Position 85
Objekte:Positionierung: Bewegung mit Geschwindigkeitsrampe ausführen 85
Objekte:Positionierung: Bewegung mit konstanter Geschwindigkeit ausführen 85

Objekte:Positionierung: Gepufferte Positionierung auf Asolutposition durchführen ... 86
Objekte:Positionierung: Gepufferte Positionierung auf Relativposition durchführen ... 86
Objekte:Positionierung: Anzahl der freien Pufferplätze auslesen ... 87
SMS ... 88
Objekt:SMS:Neue SMS zum Senden beginnen ... 88
Objekt:SMS:Text zur Sende SMS hinzufügen ... 89
Objekt:SMS:Versende SMS via GSM ... 89
Objekt:SMS:Anruf durchführen ... 90
Objekt:SMS:SMS wurde Empfangen ... 90
Objekt:SMS:Prüfe empfangenen SMS Text ... 91
Objekt:SMS:Überspringe leere Zeichen in empfangener SMS ... 91
Objekt:SMS:Hole Analogwert aus SMS ... 92
Objekt:SMS:Hole Text aus SMS ... 93
REGELUNGSTECHNIK ... 94
Objekt:Regelungstechnik: Unstetiger Hysteresen-Regler ... 94
Objekt:Regelungstechnik: Stetigähnlicher P-Regler ... 95
Objekt:Regelungstechnik:Stetiger P-Regler ... 95
Objekt:Regelungstechnik:Stetiger PI-Regler ... 96
Objekt:Regelungstechnik:Stetiger PID-Regler ... 96
DALI ... 98
Objekt:DALI:Helligkeit einer Leuchte über Kurzadresse ... 98
Objekt:DALI:Helligkeit einer Leuchte über Gruppe ... 99
Objekt:DALI:Helligkeit aller Leuchten ... 99
Objekt:DALI:Leuchte über Kurzadresse ein/aus ... 99
Objekt:DALI:Leuchtegruppe ein/aus ... 100
Objekt:DALI:Alle Leuchten ein/aus ... 100
Objekt:DALI:Szene für Kurzadresse abrufen ... 100
Objekt:DALI:Szene für Gruppe abrufen ... 101

Objekt:DALI:Szene für alle Leuchten abrufen 101
EIB/KNX KONFIGURATION 102
Objekt:EIB/KNX:Gruppenadressen konfiguieren 102
EIB/KNX VERWENDUNG 103
Objekt:EIB/KNX:Binärdaten senden 103
Objekt:EIB/KNX:Analogwerte senden 104
Objekt:EIB/KNX:Textwerte senden 104
Objekt:EIB/KNX:Binärdaten immer empfangen 104
Objekt:EIB/KNX:Analogdaten immer empfangen 105
Objekt:EIB/KNX:Textdaten immer empfangen 105
Objekt:EIB/KNX:Binärdaten Ereignis 105
Objekt:EIB/KNX:Binärdaten Ereignis mit Wert 1 (true) 106
Objekt:EIB/KNX:Binärdaten Ereignis mit Wert 0 (false) 106
Objekt:EIB/KNX:Analogdaten Ereignis 107
Objekt:EIB/KNX:Textdaten Ereignis 107
MODBUS/TCP SERVER 108
Objekt:MODBUS/TCP:Holdingregister mit Bit beschreiben 108
Objekt:MODBUS/TCP:Holdingregister mit Analogwert beschreiben 109
Objekt:MODBUS/TCP:Bitwert aus Holdingregister lesen 109
Objekt:MODBUS/TCP:Analogwert aus Holdingregister lesen 109
UDP SOCKET 111
Objekt:UDP :Nachricht mit Datenwert 1 senden 111
Objekt:UDP :Nachricht mit Datenwert 0 senden 112
Objekt:UDP :Nachricht mit Bitwert senden 112
Objekt:UDP :Nachricht mit Analogwert senden 112
Objekt:UDP :Nachricht mit Textwert senden 113
Objekt:UDP :Nachricht mit Bitwert 1 enpfangen 113
Objekt:UDP :Nachricht mit Bitwert 0 enpfangen 114
Objekt:UDP :Nachricht mit Bitwert enpfangen 114

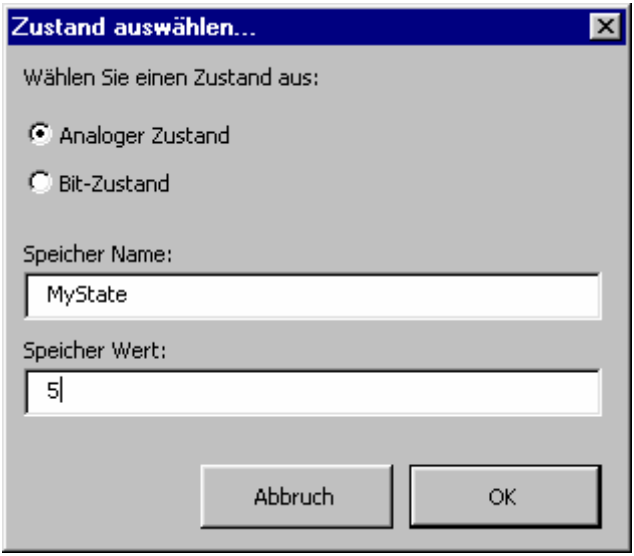
Objekt:UDP :Nachricht mit Analogwert enpfangen 115
Objekt:UDP :Nachricht mit Textwert enpfangen 115
ASCII SCHNITTSTELLEN 116
Objekt:ASCII :Schnittstelle definieren 116
Objekt:ASCII :Sendebuffer initialisieren 117
Objekt:ASCII :Text dem Sendebuffer hinzufügen 117
Objekt:ASCII :Sendebuffer senden 117
Objekt:ASCII :Empfangsbuffer auslesen 118
MODBUS/RTU MASTER 119
Objekt:MODBUS/RTU: Schnittstelle definieren 119
Objekt:MODBUS/RTU: Holdingregister vom Slave lesen 120
Objekt:MODBUS/RTU: Inputregister vom Slave lesen 120
Objekt:MODBUS/RTU: Coils vom Slave lesen 121
Objekt:MODBUS/RTU: Discrete Inputs vom Slave lesen 122
Objekt:MODBUS/RTU: Holding Register in den Slave schreiben 122
Objekt:MODBUS/RTU: Coils in den Slave schreiben 123
DEBUG 124
Debug:Symbol hinzufügen 124
Debug:Alle Symbole löschen 125
Debug:Alle Symbole updaten 125
Debug:Breakpoint setzen 125
Debug:Breakpoint löschen 125
Debug:Start mit Breakpoint 126
Debug:Monitor hinzufügen 126
Debug:Zeige Systeminformationen 127
START 128
Start:Übersetzen 128
Fehler beim Übersetzen 128
Erfolg beim Übersetzen 129
Start:Simulieren 129

Sart:Download+Start 130
Sart:SPS starten 130
Sart:SPS stoppen 130
Sart:SPS löschen 130
Start:Dialog 130
BitSpeicher lesen/schreiben 135
AnalogSpeicher lesen/schreiben 136
TextSpeicher lesen/schreiben 136
SPS nicht gefunden 136
Serielle Schnittstelle auswählen 137
SIMULATOR 139
Start des Simulators 139
Simulation:BinaryMemory 140
Simulation:AnalogMemory 141
Simulation:TextMemory 142
Simulation: Logging 143
Simulator beenden 144
KONTAKT 145

Zustände

Dieser Abschnitt beschäftigt sich mit Zuständen und deren Anwendung in Ihrer Software. Um einen neuen Zustand einzufügen, wählen Sie den Menüpunkt Fluß/Zustand an. Es erscheint folgende Eingabemaske:

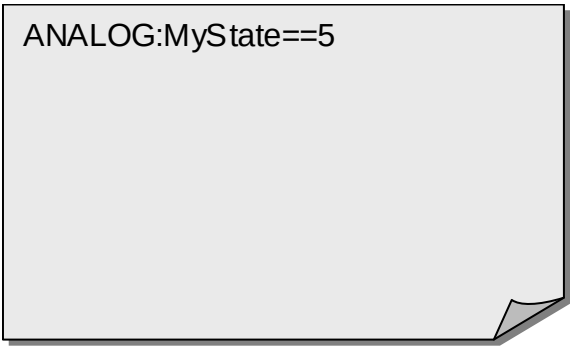
Zustand: Zustand einfügen



Hier können Sie einen analogen oder digitalen Speichernamen eingeben und einen konstanten Wert vorgeben. Zusätzlich können Sie auswählen, ob Sie sich auf einen Analogen oder binären Speicher beziehen. Nachdem Sie OK klicken wird ein Rahmen in die aktuelle Seite eingefügt, wie hier dargestellt:

Analoge Zustände

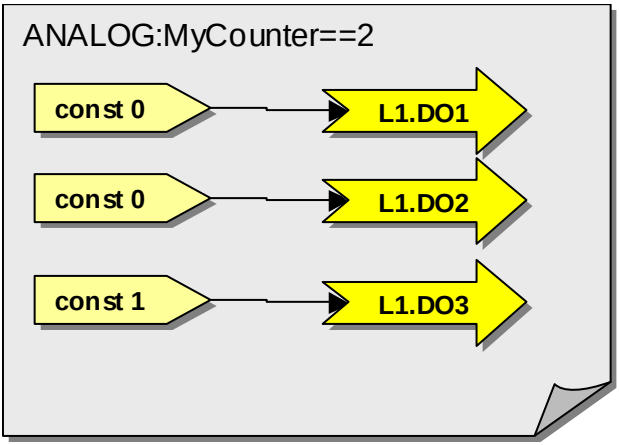
Symbol:



Datentyp: Analog

Funktion: Alle Objekte, die sich innerhalb dieses Zustandsrahmens befinden, werden nur ausgeführt, wenn der analoge Speicher MyState den Wert 5 hat. Diese Methode bewirkt innerhalb der SPS, dass die entsprechenden Programmteile wirklich übersprungen werden. Dadurch kann die Geschwindigkeit eines SPS Programmes deutlich beschleunigt werden!

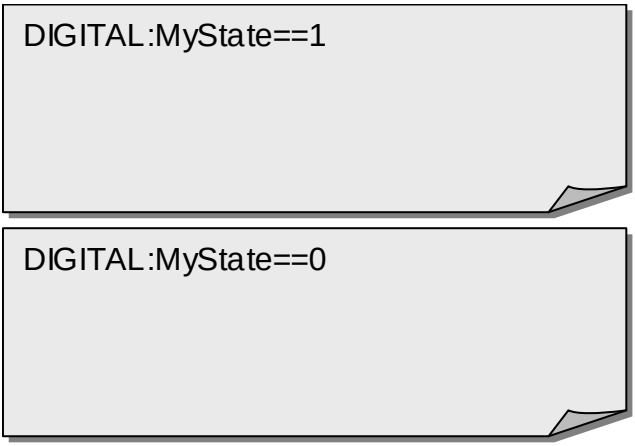
Beispiel:



Die Digitalausgäge L1.DO1, L1.DO2 und L1.DO3 werden auf die Werte 0,0,1 gesetzt, wenn der analoge Speicher MyCounter den Wert 2 hat. Ist MyCounter ungleich 2, wird kein Befehl ausgeführt.

Binäre Zustände

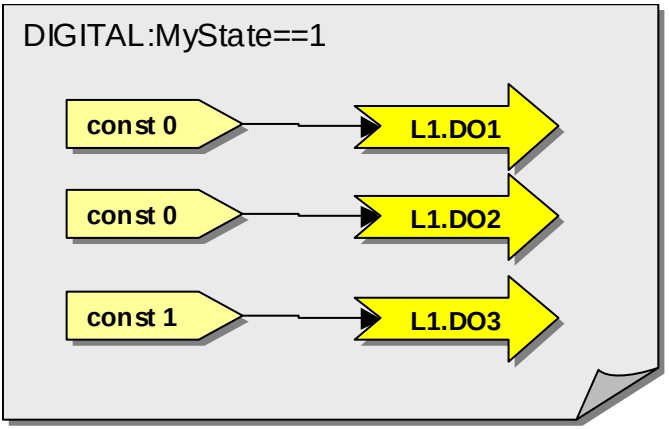
Symbol:



Datentyp: Bit

Funktion: Alle Objekte, die sich innerhalb dieses Zustandsrahmens befinden, werden nur ausgeführt, wenn der Bit-Speicher MyState den Wert 0 oder 1 hat. Diese Methode bewirkt innerhalb der SPS, dass die entsprechenden Programmteile wirklich übersprungen werden. Dadurch kann die Geschwindigkeit eines SPS Programmes deutlich beschleunigt werden!

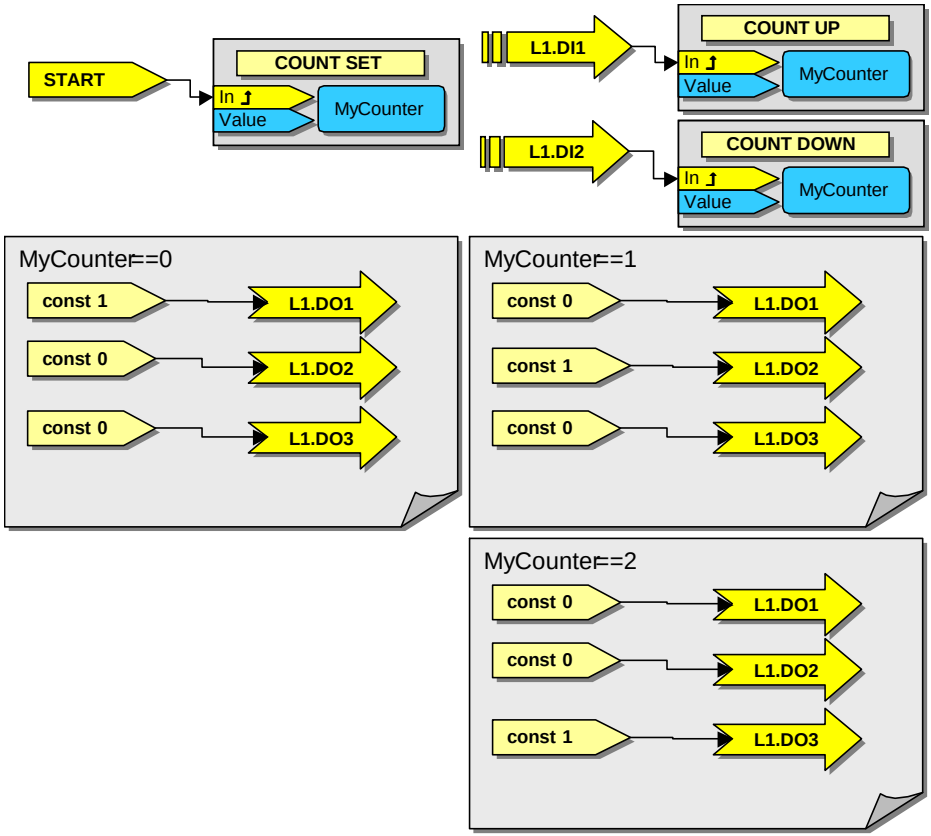
Beispiel:



Die Digitalausgäge L1.DO1, L1.DO2 und L1.DO3 werden auf die Werte 0,0,1 gesetzt, wenn der Bit-Speicher MyState den

Wert 1 hat. Ist MyState gleich Null, wird kein Befehl ausgeführt.

Beispiel: Zustandsmaschine



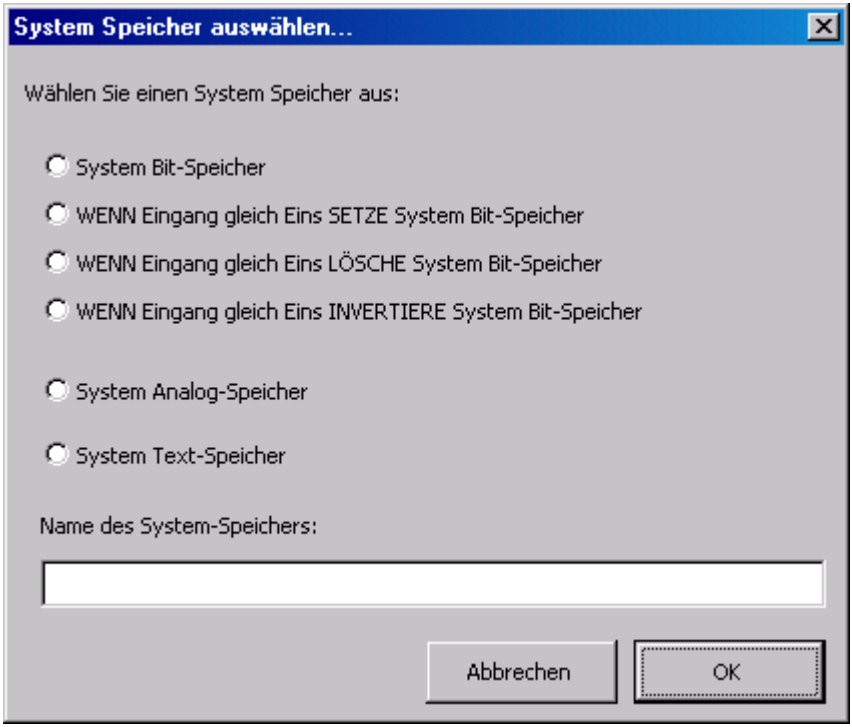
Ablauf: Bei jeder steigenden Flanke am Digitaleingang L1.DI1 wird der analoge Speicher MyCounter um eins erhöht. Bei einer steigenden Flanke am Digitaleingang L1.DI2 wird der analoge Speicher MyCounter um eins erniedrigt.

Hat MyCounter den Wert 0, so wird nur der Ausgang L1.DO1 gesetzt. Bei MyCounter=1 wird der Ausgang L1.DO2 aktiviert,

bei MyCounter=3 wird der Ausgang L1.DO3 aktiviert. Bei allen übrigen Werten von MyCounter findet keine Aktion statt.

System

In diesem Abschnitt befinden sich Befehle, die sich mit den Systemvariablen auseinandersetzen. Dazu wählen Sie den Menüpunkt Fluss/System aus. Es erscheint folgende Auswahl:

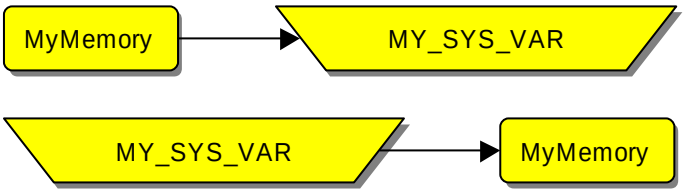


System: System Bit-Speicher

Symbol:

Datentyp: Bit

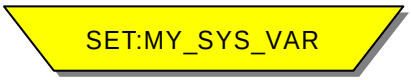
Funktion: Greift auf einen Bitspeicher des Systems zu. Der Name ist fix im System-Bitspeicher verankert und muss daher exakt dem Namen des vorhandenen Speichernamens entsprechen.



Beispiel:

Hier wird der fiktive System-Bitspeicher auf den Wert von MyMemory gesetzt und eine Zeile darunter wird MyMemory mit dem Wert von MY_SYS_VAR beschrieben.

System: WENN Eingang gleich Eins SETZE Bit-Speicher

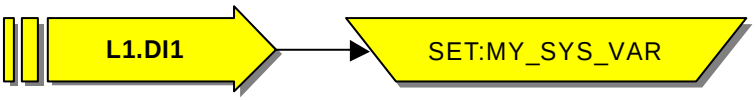


Symbol:

Datentyp: Bit

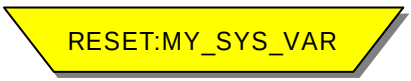
Funktion: Liegt am Eingang dieses System-Bitspeichers eine 1 an, so wird MY_SYS_VAR auf eins gesetzt.

Beispiel:



Wird der Digitaleingang L1.DI1 betätigt, so wird der System-Bitspeicher MY_SYS_VAR auf eins gesetzt.

System: WENN Eingang gleich Eins LÖSCHE Bit-Speicher

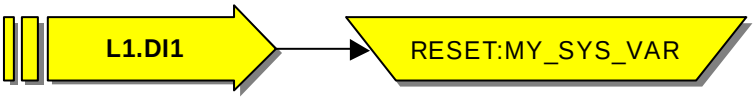


Symbol:

Datentyp: Bit

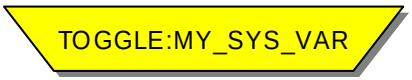
Funktion: Liegt am Eingang dieses System-Bitspeichers eine 1 an, so wird MY_SYS_VAR gelöscht.

Beispiel:



Wird der Digitaleingang L1.DI1 betätigt, so wird der System-Bitspeicher MY_SYS_VAR gelöscht.

System: WENN Eingang gleich Eins INVERTIERE Bit-Sp.

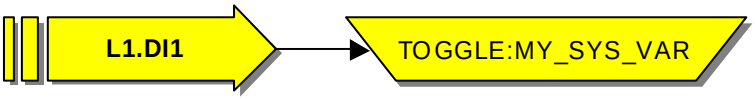


Symbol:

Datentyp: Bit

Funktion: Liegt am Eingang dieses System-Bitspeichers eine 1 an, so wird der aktuelle Inhalt von MY_SYS_VAR invertiert.

Beispiel:



Wird der Digitaleingang L1.DI1 betätigt, so wird der aktuelle Inhalt des System-Bitspeichers MY_SYS_VAR invertiert.

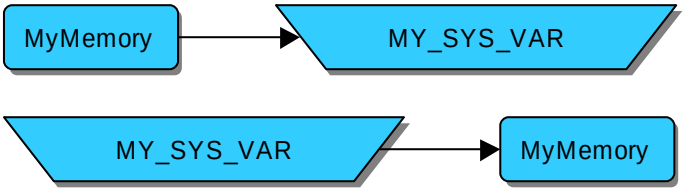
System: System Analog-Speicher

Symbol:

Datentyp: Analog

Funktion: Greift auf einen Bitspeicher des Systems zu. Der Name ist fix im System-Analogspeicher verankert und muss daher exakt dem Namen des vorhandenen Speichernamens entsprechen.

Beispiel:



Hier wird der fiktive System-Analogspeicher auf den Wert von MyMemory gesetzt und eine Zeile darunter wird MyMemory mit dem Wert von MY_SYS_VAR beschrieben.

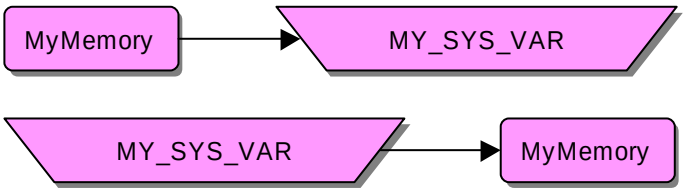
System: System Text-Speicher

Symbol:

Datentyp: Text

Funktion: Greift auf einen Textspeicher des Systems zu. Der Name ist fix im System-Textspeicher verankert und muss daher exakt dem Namen des vorhandenen Speichernamens entsprechen.

Beispiel:



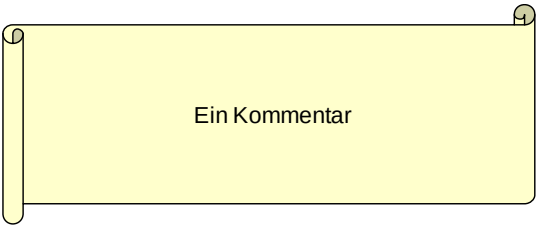
Hier wird der fiktive System-Textspeicher auf den Wert von MyMemory gesetzt und eine Zeile darunter wird MyMemory mit dem Wert von MY_SYS_VAR beschrieben.

Kommentare + Symbolische Gruppen

Prinzipiell ignoriert RESI'S GRAPH beim Übersetzen des Programms alle PowerPoint Objekte, die nicht über die RESI'S GRAPH-Menüleiste eingefügt wurden. Ein spezielles Kommentarobjekt stellt aber auch die RESI'S GRAPH-Leiste zur Verfügung: Wählen Sie dazu den Menüpunkt Fluß/Kommentar

Kommentare einfügen

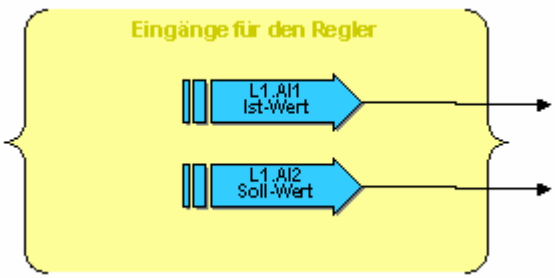
Symbol:



Funktion: Fügt einen Kommentar in die aktuelle Programmseite ein. Klicken Sie auf den Text und geben Sie Ihren gewünschten Kommentar ein! Sie können beliebig viele Kommentare auf einer Seite einfügen. Diese werden bei der Programmerstellung ignoriert.

Symbolische Gruppe erstellen

Symbol:



Funktion: Fügt einen Hintergrund mit in die aktuelle Programmseite ein. Klicken Sie auf den Text und geben Sie Ihren gewünschten

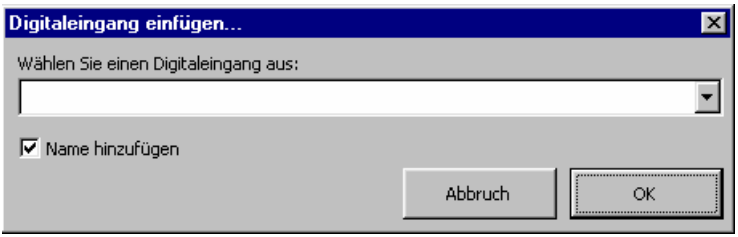
Titel für die Gruppe ein! Ziehen Sie die Objekte, die eine Gruppe bilden sollen, einfach in das Feld. Sie können beliebig viele symbolische Gruppen auf einer Seite erstellen. Dies ändert nichts an der Programmerstellung.

I/O

Unter der Gruppe der I/O verbergen sich alle Funktionen, die sich mit dem Einfügen von digitalen oder analogen Ein- oder Ausgängen beschäftigen. Folgendes Kapitel erklärt den Einsatz der Ein/Ausgänge von RESI'S GRAPH.

I/O: Digitaleingäne

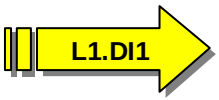
Wählen Sie den Menüpunkt I/O/Digitaleingänge an. Es erscheint folgende Auswahl:



Aus der Liste können Sie einen Digitaleingang auswählen und mit OK bestätigen. Durch das aktivieren von Name hinzufügen fügt man den Variaben Name im Symbol ein. Ist Name hinzufügen nicht aktiv so wird nur der Digitale Input im Symbol angezeigt.

WICHTIG Sie können erst dann einen Digitaleingang wählen, wenn unter der Seite CONFIG eine ordentliche Anlagenkonfiguration erstellt wurde!

Symbol ohne Name hinzufügen:

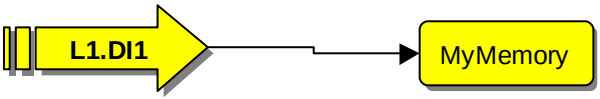


Symbol mit aktivierten Name hinzufügen:

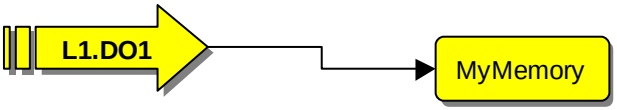
Datentyp: Bit

Funktion: Diese Funktion fügt einen Digitaleingang ein. Digitaleingänge können Anfangspunkte einer Verbindung sein. In der Liste der Digitaleingänge finden Sie auch alle Digitalausgänge, um den aktuellen Status der Ausgänge weiterzuverarbeiten.

Beispiel:



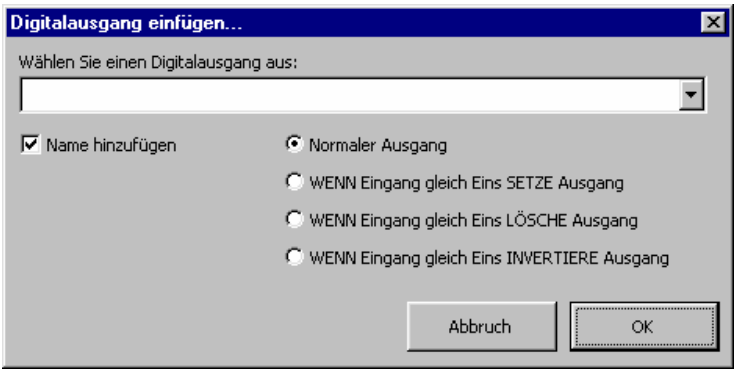
Der aktuelle Zustand des Digitaleinganges L1.DI1 wird in den Bitspeicher MyMemory übertragen.



Der aktuelle Zustand des Digitalausganges L1.DO1 wird in den Bitspeicher MyMemory übertragen.

I/O: Digitalausgänge

Wählen Sie den Menüpunkt I/O/Digitalausgänge an. Es erscheint folgende Auswahl:



Aus der Liste können Sie einen Digitaleingang auswählen und mit OK bestätigen. Durch das aktivieren von Name hinzufügen fügt man den Variablen Name im Symbol ein. Ist Name hinzufügen nicht aktiv so wird nur der Digitale Output im Symbol angezeigt.

Zusätzlich können Sie zwischen vier Funktionen des Ausgangs wählen:

Normal output: Der Ausgang nimmt immer den Wert der ankommenden Verbindung an.

IF input is high SET output: Der Ausgang wird auf High gesetzt, wenn die Verbindung 1 ist. Ist die Verbindung 0, so bleibt der Wert des Ausgangs unverändert.

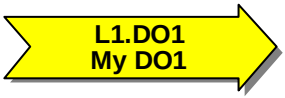
IF input is high RESET output: Der Ausgang wird auf Low gesetzt, wenn die Verbindung 1 ist. Ist die Verbindung 0, so bleibt der Wert des Ausgangs unverändert.

IF input is high TOGGLE output: Der Ausgang wird invertiert, solange die Verbindung 1 ist. Ist die Verbindung 0, so bleibt der Wert des Ausgangs unverändert.

WICHTIG Sie können erst dann einen Digitalausgang wählen, wenn unter der Seite CONFIG eine ordentliche Anlagenkonfiguration erstellt wurde!

Symbol ohne Name hinzufügen:

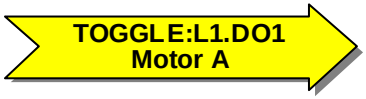
Symbol mit aktivierten Name hinzufügen:



Symbol für Funktion SETZE:

Symbol für Funktion LÖSCHEN:

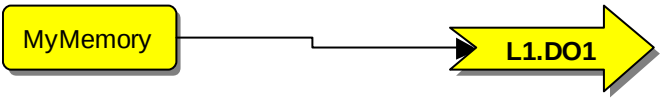
Symbol für Funktion INVERTIEREN:



Datentyp: Bit

Funktion: Diese Funktion fügt einen Digitalausgang ein. Digitalausgänge stellen immer den Endpunkt einer Verbindung dar. Um den aktuellen Zustand eines Digitalausganges abzufragen, müssen sie aus der Liste der Digitaleingänge den Ausgang auswählen.

Beispiel:



Der aktuelle Zustand des Bitspeichers MyMemory wird auf den Digitalausgang L1.DO1 übertragen.

I/O: Analogeingänge

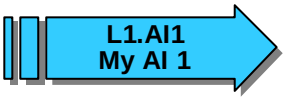
Wählen Sie den Menüpunkt I/O/Analogeingänge an. Es erscheint folgende Auswahl:

Aus der Liste können Sie einen Analogeingang auswählen und mit OK bestätigen. Durch das aktivieren von Name hinzufügen fügt man den Variablen Name im Symbol ein. Ist Name hinzufügen nicht aktiv so wird nur der Analog Input im Symbol angezeigt.

WICHTIG Sie können erst dann einen Analogeingang wählen, wenn unter der Seite KONFIGURATION eine ordentliche Anlagenkonfiguration erstellt wurde!

Symbol ohne Name hinzufügen:

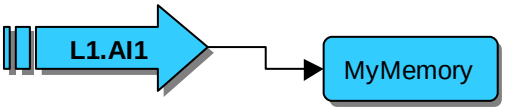
Symbol mit aktivierten Name hinzufügen:



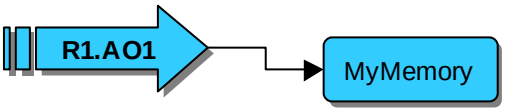
Datentyp: Analog

Funktion: Diese Funktion fügt einen Analogeingang ein. Analogeingänge stellen immer den Anfang einer Verbindung dar. Alle Analogausgänge stehen zum Rücklesen ebenfalls als Analogeingänge zur Verfügung. Prinzipiell werden alle Analogsignale als Werte zwischen 0.000 und 100.000 dargestellt und stellen somit einen Prozentwert zwischen 0% und 100% dar, unabhängig von der realen Beschaltung!

Beispiel:



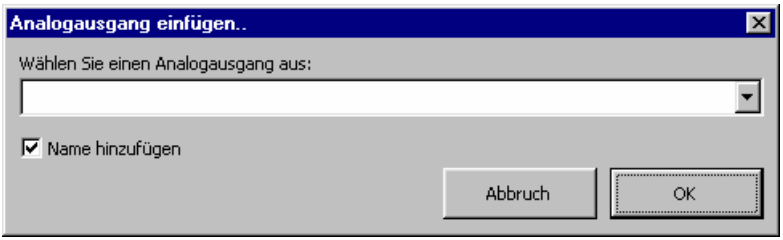
Der aktuelle Zustand des Analogeinganges L1.AI1 wird in den analogen Speicher MyMemory übertragen.



Der aktuelle Zustand des Analogausganges R1.AO1 wird in den analogen Speicher MyMemory abgelegt.

I/O: Analogausgänge

Wählen Sie den Menüpunkt I/O/Analogausgänge an. Es erscheint folgende Auswahl:

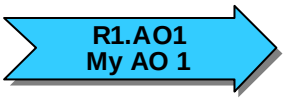


Aus der Liste können Sie einen Analogausgang auswählen und mit OK bestätigen. Durch das aktivieren von Name hinzufügen fügt man den Variablen Name im Symbol ein. Ist Name hinzufügen nicht aktiv so wird nur der Analog Output im Symbol angezeigt.

WICHTIG Sie können erst dann einen Analogausgang wählen, wenn unter der Seite KONFIGURATION eine ordentliche Anlagenkonfiguration erstellt wurde!

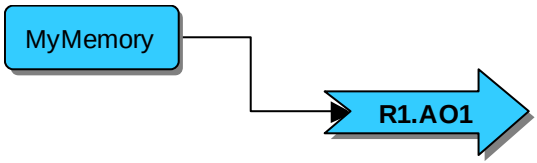
Symbol ohne Name hinzufügen:

Symbol mit aktivierten Name hinzufügen:



Datentyp: Analog

Funktion: Diese Funktion fügt einen Analogausgang ein. Analogausgänge stellen immer das Ende einer Verbindung dar. Alle Analogausgänge stehen zum Rücklesen ebenfalls als Analogeingänge zur Verfügung. Prinzipiell werden alle Analogsignale als Werte zwischen 0.000 und 100.000 dargestellt und stellen somit einen Prozentwert zwischen 0% und 100% dar, unabhängig von der realen Beschaltung!

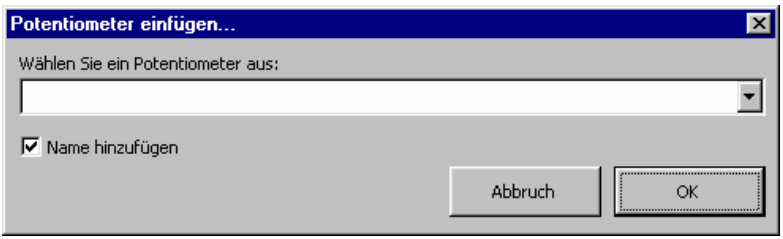


Beispiel:

Der aktuelle Wert des analogen Speichers MyMemory wird auf den Analogausgang R1.AO1 übertragen.

I/O: Potentiometer

Wählen Sie den Menüpunkt I/O/Potentiometer an. Es erscheint folgende Auswahl:



Aus der Liste können Sie einen Potentiometer auswählen und mit OK bestätigen. Durch das aktivieren von Name hinzufügen fügt man den Variablen Name im Symbol ein. Ist Name hinzufügen nicht aktiv so wird nur der Poti im Symbol angezeigt.

WICHTIG Sie können erst dann einen Potentiometer wählen, wenn unter der Seite KONFIGURATION eine ordentliche Anlagenkonfiguration erstellt wurde!

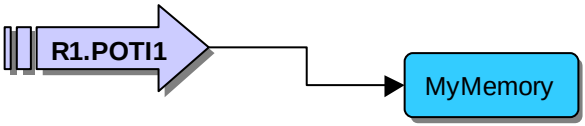
Symbol ohne Name hinzufügen:

Symbol mit aktivierten Name hinzufügen:

Datentyp: Analog

Funktion: Diese Funktion fügt ein Potentiometer ein. Potentiometer stellen immer den Anfang einer Verbindung dar. Prinzipiell werden alle Analogsignale als Werte zwischen 0.000 und 100.000 dargestellt und stellen somit einen Prozentwert zwischen 0% und 100% dar, unabhängig von der realen Beschaltung!

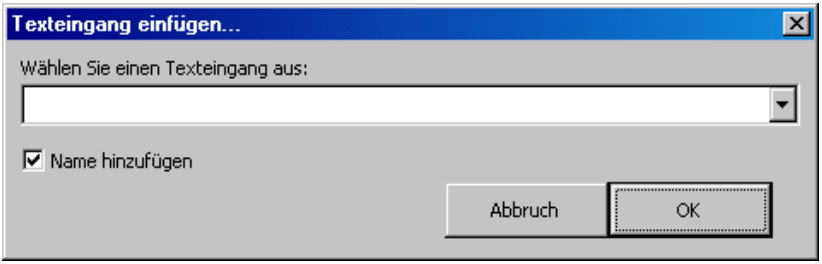
Beispiel:



Der aktuelle Potentiometerwert R1.POTI1 wird im Speicher MyMemory abgelegt.

I/O: Texteingänge

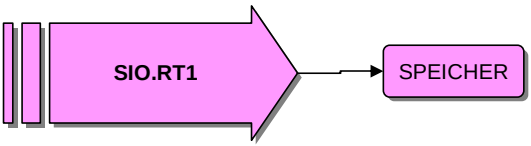
Wählen Sie den Menüpunkt I/O/Texteingänge an. Es erscheint folgende Auswahl:



Aus der Liste können Sie einen Texteingang auswählen und mit OK bestätigen. Durch das aktivieren von Name hinzufügen fügt man den Variablen Name im Symbol ein. Ist Name hinzufügen nicht aktiv so wird nur der Text Input im Symbol angezeigt.

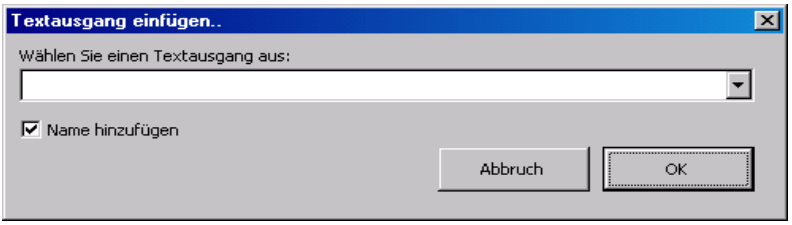
WICHTIG Sie können erst dann einen Texteingang wählen, wenn unter der Seite KONFIGURATION eine ordentliche Anlagenkonfiguration erstellt wurde!

BEISPIELE



I/O: Textausgänge

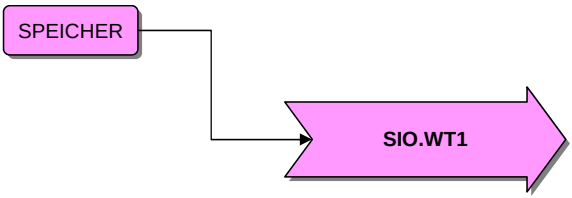
Wählen Sie den Menüpunkt I/O/Textausgänge an. Es erscheint folgende Auswahl:



Aus der Liste können Sie einen Textausgang auswählen und mit OK bestätigen. Durch das aktivieren von Name hinzufügen fügt man den Variablen Name im Symbol ein. Ist Name hinzufügen nicht aktiv so wird nur der Text Output im Symbol angezeigt.

WICHTIG Sie können erst dann einen Analogausgang wählen, wenn unter der Seite KONFIGURATION eine ordentliche Anlagenkonfiguration erstellt wurde!

BEISPIELE

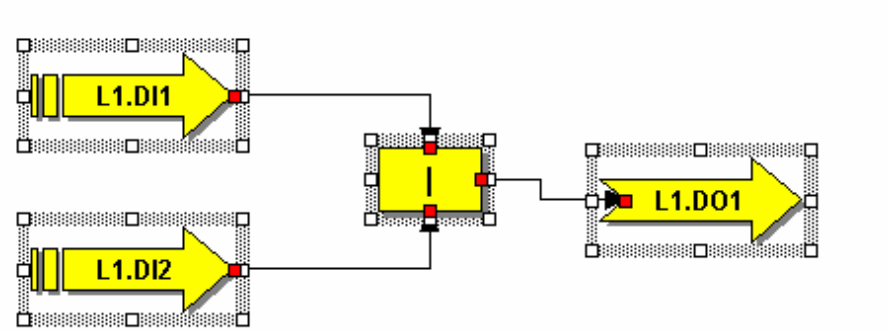


Gruppen

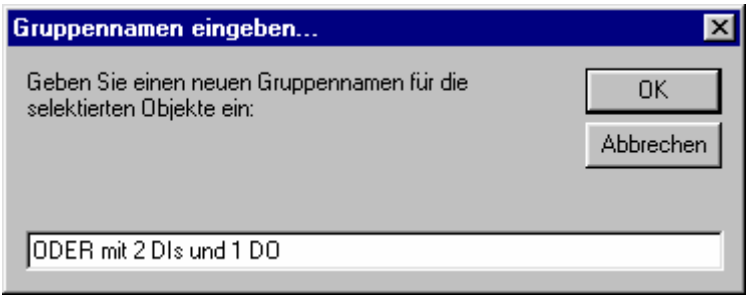
Gruppen sind häufig verwendete Anordnungen von RESI'S GRAPH Elementen und Verbindungen. Jeder Gruppe kann ein Name zugeordnet werden. Über diesen Namen kann man jederzeit die Elemente der Gruppe in die aktuelle Seite einfügen. Gruppen sind nur innerhalb eines Projektes verfügbar. Man kann Gruppen nicht von einem Projekt in ein anderes übernehmen!

Gruppen exportieren

Markieren Sie eines oder mehrere Objekte, die Sie als Gruppe speichern wollen:



Danach wählen Sie den Menüpunkt Group/Export an. Es erscheint folgendes Eingabefenster:

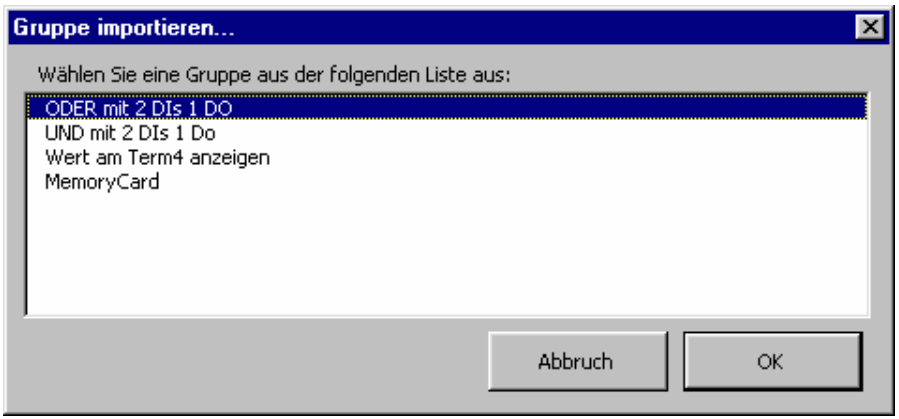

Gruppennamen eingeben...
Geben Sie einen neuen Gruppennamen für die selektierten Objekte ein:
OK
Abbrechen
ODER mit 2 DIs und 1 DO

Geben Sie nun eine eindeutige Bezeichnung für Ihre Gruppe ein und drücken Sie OK. Schon ist Ihre Gruppe exportiert.

Gruppen importieren

Um eine bereits gespeicherte Gruppe wieder zu importieren, gehen Sie wie folgt vor:

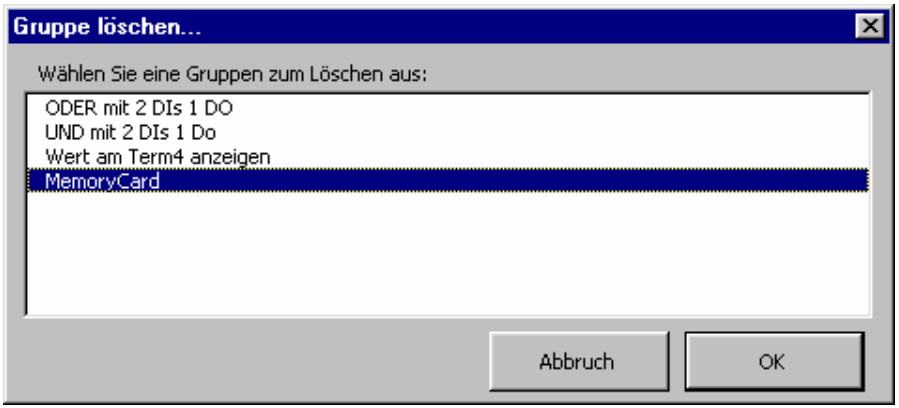
Wählen Sie den Menüpunkt Gruppe/importieren an. Es erscheint folgende Liste:



Wählen Sie aus der Liste die Gruppe, die Sie importieren wollen und drücken Sie OK. Alle Elemente der Gruppe werden in die aktuelle Seite eingefügt und stehen dort wieder als Einzelelemente zur Verfügung!

Gruppen löschen

Um eine Gruppe zu löschen, wählt man den Menüpunkt Gruppe/Löschen. Es erscheint eine Auswahl mit allen verfügbaren Gruppen:

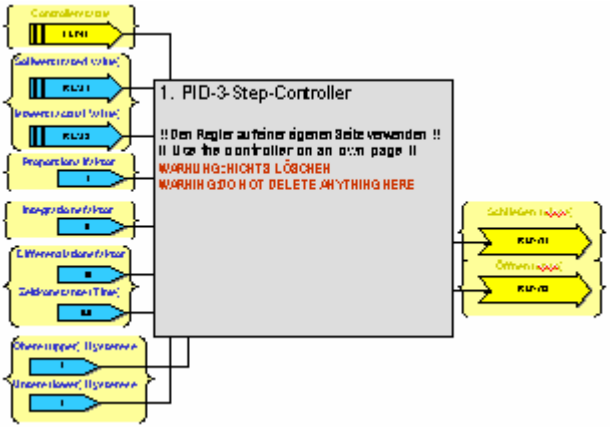


Wählen Sie die Gruppe aus die Sie löschen wollen und bestätigen Sie mit OK.

Regler einfügen

Wählen Sie aus der Liste den Regler, den Sie verwenden wollen und drücken Sie OK. Alle Elemente des Reglers werden in die aktuelle Seite eingefügt und stehen dort wieder als Einzelelemente zur Verfügung!

!WICHTIG!: Jeder Regler darf nur einmal auf einer eigenen Seite verwendet werden, das heißt, wenn Sie aus unseren SIGRAPH-Angebot an Standart-Reglern wählen, stehen Ihnen für ein Programm bis zu 7 vorprogrammierte Reglertools zur Verfügung.

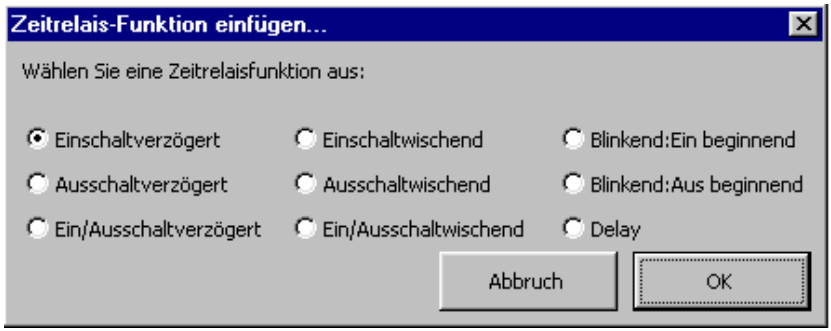

1. PID-3-Step-Controller
!! Den Regler auf einer eigenen Seite verwenden !!
!! Use the controller on an own page !!
WARNUNG:NICHTS LÖSCHEN
WARNING:DO NOT DELETE ANYTHING HERE

Objekte

Unter dem Menüpunkt Objekte gibt es eine Reihe von Sonderfunktionen, die in RESI'S GRAPH verwendet werden können.

Objekte:Zeitrelais

Wenn Sie den Menüpunkt Timer anwählen, wird folgende Auswahl sichtbar:

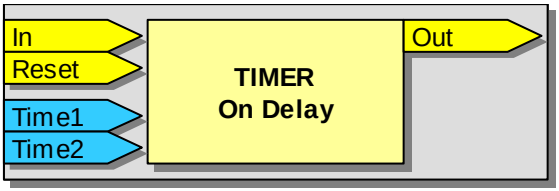


Wählen Sie nun einen der gewünschten Timermodule an:

- Einschaltverzögerung
- Ausschaltverzögerung
- Ein/Ausschaltverzögerung
- Einschaltwischend
- Ausschaltwischend
- Ein/Ausschaltwischend
- Blinken:Ein beginnend
- Blinken:Aus beginnend
- Delay: einfache Verzögerung

Nach dem OK-Klick erscheint folgendes Symbol:

Symbol:



Datentyp:

In	Bit
Reset	Bit
Out	Bit
Time1	Analog
Time2	Analog

Eingänge: In: Wenn dieser Eingang aktiv ist, wird die gewählte Zeitfunktion gestartet.

Reset: Sobald dieser Eingang high ist, wird der Ausgang auf 0 gesetzt

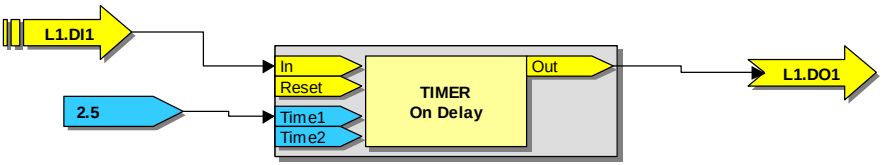
Out: Der Ausgang der Zeitfunktion. Je nach eingestellter Funktion ändert sich hier der Zustand

Time1: Die erste Zeit des Timers in Sekunden.

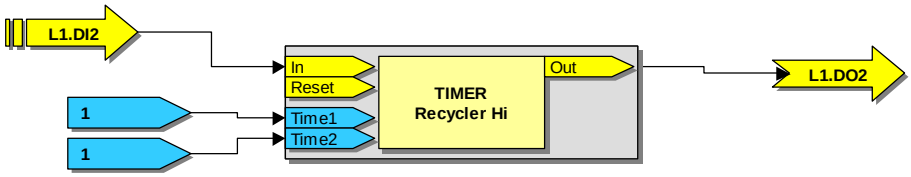
Time2: Die zweite Zeit des Timers in Sekunden. Diese wird nur bei den Zeitfunktionen Recycler high first und Recycler low fisr benötigt.

Zeitbasis: Alle Timerfunktionen basieren auf einer Zeitauflösung von 100ms!

Beispiel:

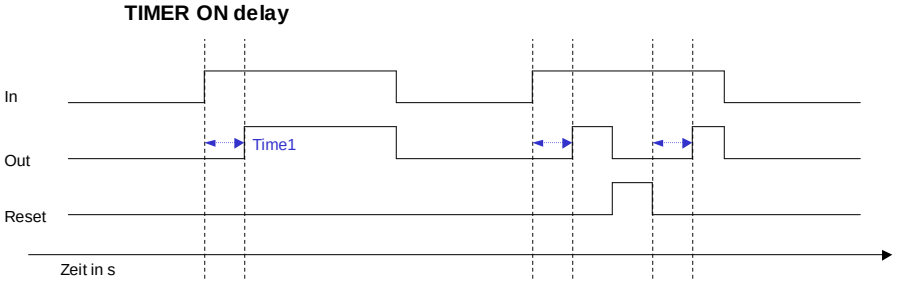


In diesem Beispiel wird 2.5s nach Aktivierung des Digitaleinganges L1.DI1 der Digitalausgang L1.DO1 aktiv

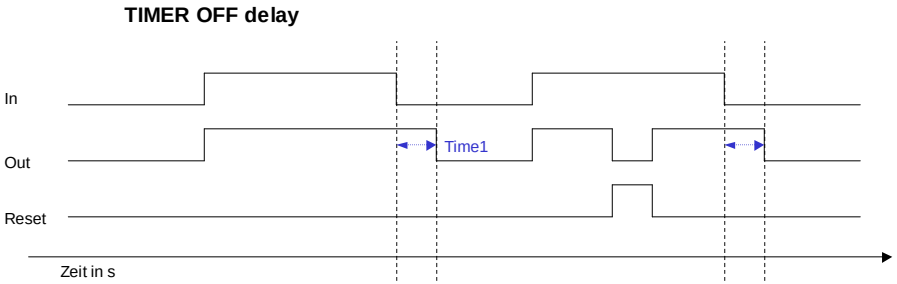


In diesem Beispiel wird der Digitalausgang L1.DO2 im 1s Takt blinken, solange der Digitaleingang L1.DI2 aktiv ist.

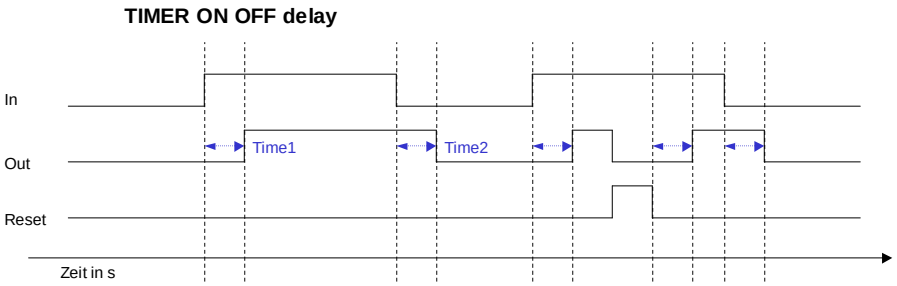
Objekte:Zeitrelais: Einschaltverzögerung



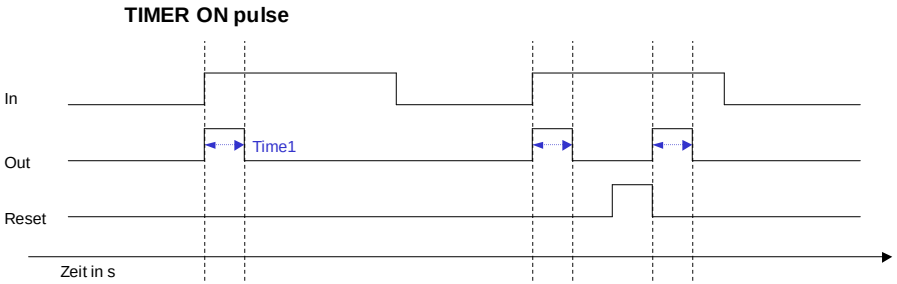
Objekte:Zeitrelais: Ausschaltverzögerung



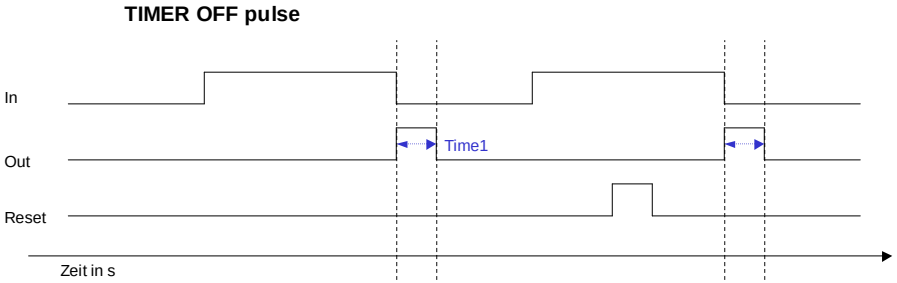
Objekte:Zeitrelais: Ein/Ausschaltverzögerung



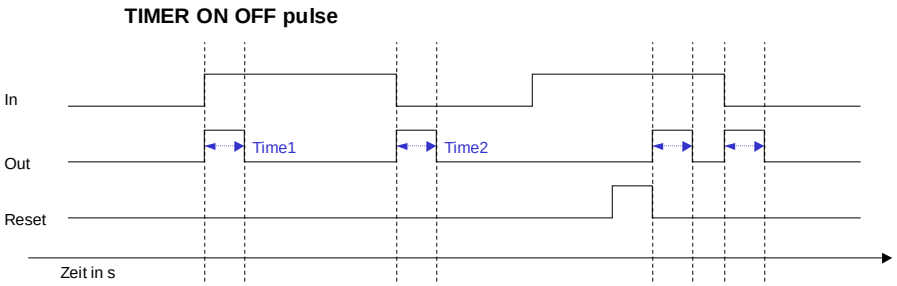
Objekte:Zeitrelais: Einschaltwischend



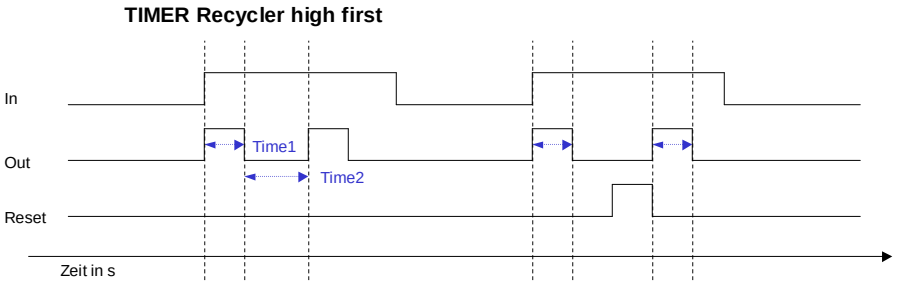
Objekte:Zeitrelais: Ausschaltwischend



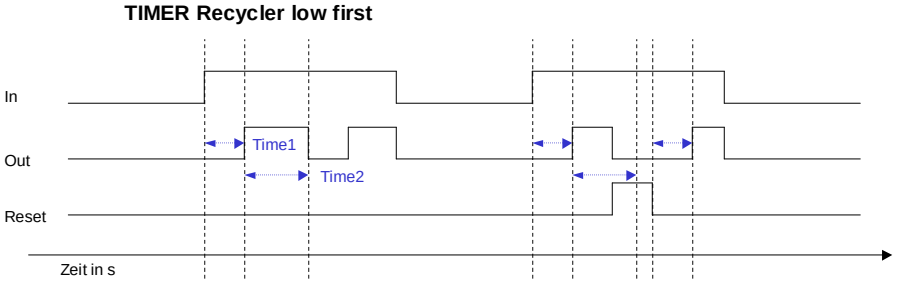
Objekte:Zeitrelais: Ein/Ausschaltwischend



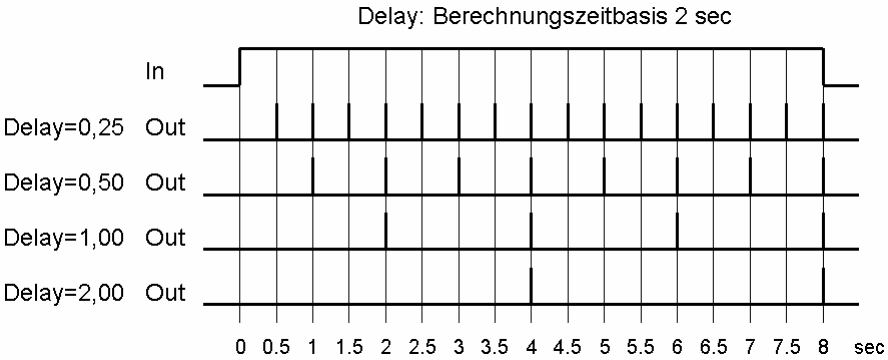
Objekte:Zeitrelais: Blinken ein beginnend



Objekte:Zeitrelais: Blinken aus beginnend



Objekte: Zeitrelais: Delay

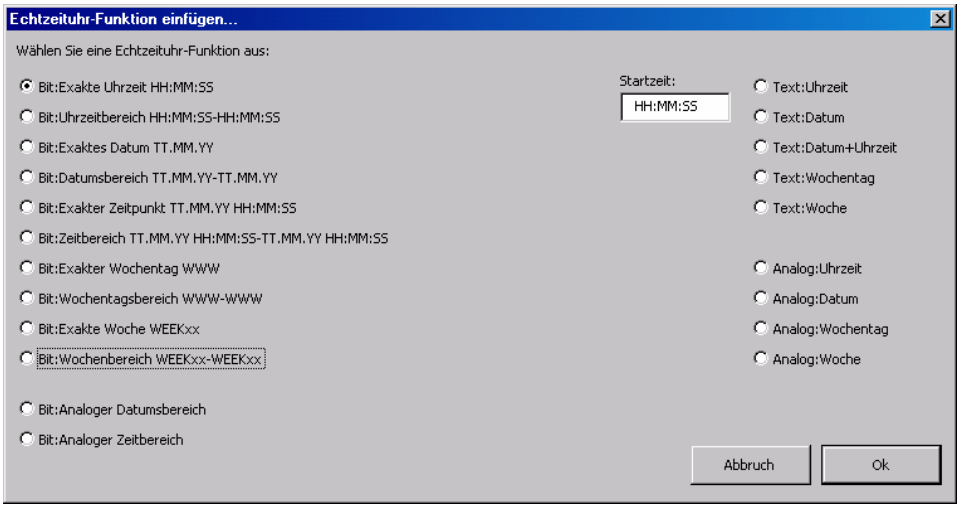

Delay: Berechnungszeitbasis 2 sec
In
Delay=0,25 Out
Delay=0,50 Out
Delay=1,00 Out
Delay=2,00 Out
0 0.5 1 1.5 2 2.5 3 3.5 4 4.5 5 5.5 6 6.5 7 7.5 8 sec

Echtzeituhr

Der folgende Abschnitt beschäftigt sich mit den Funktionen der integrierten Echtzeituhr.

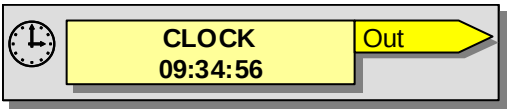
HINWEIS: Die Echtzeituhr steht nicht auf allen Modulen zur Verfügung. Bitte sehen Sie bei den Eigenschaften der Controller nach, ob die Uhr unterstützt wird!

Um Uhrenfunktionen einzubinden, wählen Sie den Menüpunkt Objekte/Echtzeituhr. Es erscheint folgende Auswahl:



Objekte:Uhr:Exakte Uhrzeit

Symbol:



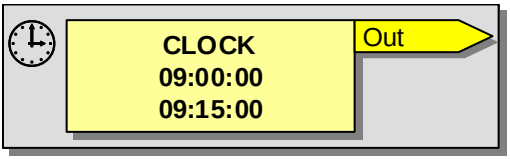
Datentyp: Out Bit

Eingabefeld: Starttime Zeitpunkt im 24h Format HH:MM:SS

Funktion: Diese Funktion vergleicht die aktuelle Uhrzeit des Moduls mit der voreingestellten Uhrzeit. Ist der Zeitpunkt erreicht, wird der Ausgang Out aktiviert.

Objekte:Uhr:Uhrzeitbereich

Symbol:



Datentyp: Out Bit

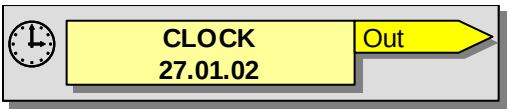
Eingabefeld: Starttime Anfangszeitpunkt im 24h Format HH:MM:SS

Endtime Endzeitpunkt im 24h Format HH:MM:SS

Funktion: Diese Funktion vergleicht die aktuelle Uhrzeit des Moduls mit dem definierten Uhrzeitbereich. Ist die aktuelle Uhrzeit zwischen der Startzeit und der Endzeit, so ist der Ausgang Out aktiv, sonst bleibt dieser 0.

Objekte:Uhr:Exaktes Datum

Symbol:



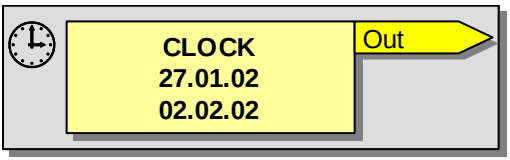
Datentyp: Out Bit

Eingabefeld: Startdate Tag im Format TT.MM.YY

Funktion: Diese Funktion vergleicht das aktuelle Datum des Moduls mit dem voreingestellten Datum. Solange das Datum übereinstimmt, wird der Ausgang Out aktiviert.

Objekte:Uhr:Datumsbereich

Symbol:



Datentyp: Out Bit

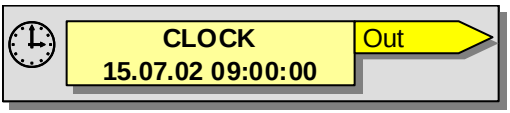
Eingabefeld: Startdatum Starttag im Format TT.MM.YY

Enddatum Endtag im Format TT.MM.YY

Funktion: Diese Funktion vergleicht das aktuelle Datum des Moduls mit dem voreingestellten Datumsbereich. Solange das aktuelle Datum innerhalb des eingestellten Datumsbereichs liegt, wird der Ausgang Out aktiviert.

Objekte:Uhr:Exakter Zeitpunkt

Symbol:



Datentyp: Out Bit

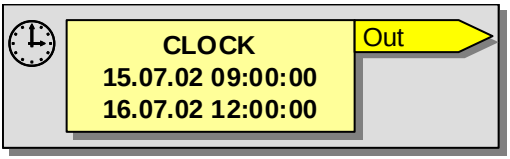
Eingabefeld: Startdatum Datum im Format TT.MM.YY

Startzeit Uhrzeit im 24h-Format HH:MM:SS

Funktion: Diese Funktion vergleicht das aktuelle Datum und die aktuelle Uhrzeit des Moduls mit den voreingestellten Datum und der voreingestellten Uhrzeit. Stimmt beides überein, so wird der Ausgang Out aktiviert.

Objekte:Uhr:Zeitbereich

Symbol:



Datentyp: Out Bit

Eingabefeld: Startdatum Startdatum im Format TT.MM.YY

Startzeit Startuhrzeit im 24h-Format HH:MM:SS

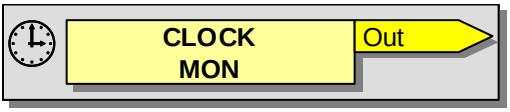
Enddatum Enddatum im Format TT.MM.YY

Endzeit Enduhrzeit im 24h-Format HH:MM:SS

Funktion: Diese Funktion vergleicht die aktuelle Uhrzeit und das aktuelle Datum des Moduls mit dem angegebenen Zeitbereich. Nur wenn die aktuelle Uhrzeit innerhalb des spezifizierten Zeitbereichs liegt, ist der Ausgang Out aktiv.

Objekte:Uhr:Exakter Wochentag

Symbol:



Datentyp: Out Bit

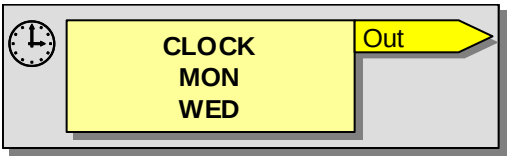
Eingabefeld: Starttag Wochentag in englischer Schreibweise:

MON,TUE,WED,THU,FRI,SAT,SUN

Funktion: Diese Funktion vergleicht den aktuelle Wochentag des Moduls mit dem angegebenen Wochentag. Stimmt der Wochentag überein, so ist der Ausgang Out aktiv.

Objekte:Uhr:Wochentagsbereich

Symbol:



Datentyp: Out Bit

Eingabefeld: Starttag Wochentag in englischer Schreibweise:

MON,TUE,WED,THU,FRI,SAT,SUN

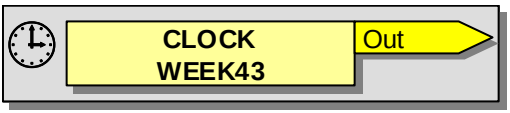
Endtag Wochentag in englischer Schreibweise:

MON,TUE,WED,THU,FRI,SAT,SUN

Funktion: Diese Funktion vergleicht den aktuellen Wochentag des Moduls mit dem angegebenen Wochentagsbereich. Liegt der aktuelle Wochentag innerhalb der spezifizierten Wochentage, so ist der Ausgang Out aktiv.

Objekte:Uhr:Exakte Woche

Symbol:



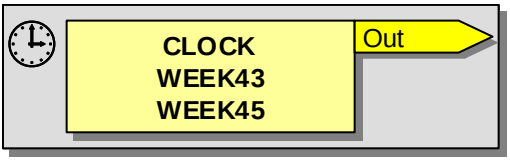
Datentyp: Out Bit

Eingabefeld: Startwoche Kalenderwoche im Format WEEKxx

Funktion: Diese Funktion vergleicht die aktuelle Kalenderwoche des Moduls mit der angegebenen Kalenderwoche. Stimmt die Kalenderwoche überein, so ist der Ausgang Out aktiv.

Objekte:Uhr:Wochenbereich

Symbol:



Datentyp: Out Bit

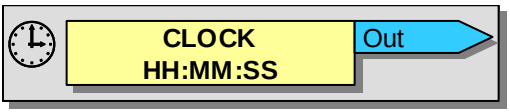
Eingabefeld: Startwoche Kalenderwoche im Format WEEKxx

Endwoche Endkalenderwoche im Format WEEKxx

Funktion: Diese Funktion vergleicht die aktuelle Kalenderwoche des Moduls mit dem angegebenen Kalenderwochenbereich. Liegt die aktuelle Kalenderwoche innerhalb des angegebenen Bereichs, so ist der Ausgang Out aktiv.

Objekte:Uhr:Analog:Uhrzeit

Symbol:



Datentyp: Out Analog

Funktion: Diese Funktion liefert die aktuelle Uhrzeit des Moduls als Analogwert. Der Analogwert ist wie folgt aufgebaut:

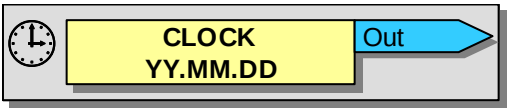
0x00HHMMSS beschreibt eine 24 Bit Hexadezimalzahl.

Die Kodierung der Gruppen HH,MM,SS erfolgt als Dezimalzahl.

Beispiel: Die Uhrzeit 14:57:36 liefert als Analogwert 0x000E3924

Objekte:Uhr:Analog:Datum

Symbol:



Datentyp: Out Analog

Funktion: Diese Funktion liefert das aktuelle Datum des Moduls als Analogwert. Der Analogwert ist wie folgt aufgebaut:

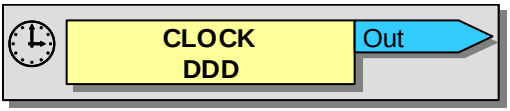
0x00YYMMDD beschreibt eine 24 Bit Hexadezimalzahl.

Die Kodierung der Gruppen YY,MM,DD erfolgt als Dezimalzahl.

Beispiel: Das Datum 16.05.02 liefert als Analogwert 0x00020510

Objekte:Uhr:Analog:Wochentag

Symbol:



Datentyp: Out Analog

Funktion: Diese Funktion liefert den aktuellen Wochentag des Moduls als Analogwert. Der Analogwert ist wie folgt aufgebaut:

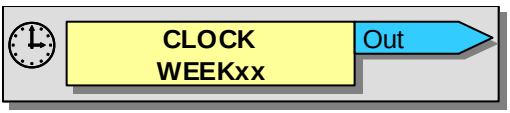
0x0000000D beschreibt eine 4 Bit Hexadezimalzahl.

Die Kodierung der Gruppe D erfolgt als Dezimalzahl. Der Montag stellt den Wert 0 dar, Dienstag 1, usw.

Beispiel: Der Wochentag Donnerstag (THU) liefert 0x00000003 als Ergebnis.

Objekte:Uhr:Analog:Woche

Symbol:



Datentyp: Out Analog

Funktion: Diese Funktion liefert die aktuelle Kalenderwoche des Moduls als Analogwert. Der Analogwert ist wie folgt aufgebaut:

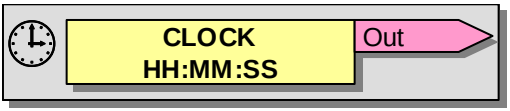
0x000000WW beschreibt eine 8 Bit Hexadezimalzahl.

Die Kodierung der Gruppe WW erfolgt als Dezimalzahl.

Beispiel: Die Kalenderwoche 17 liefert als Ergebnis 0x00000011.

Objekte:Uhr:Text:Uhrzeit

Symbol:

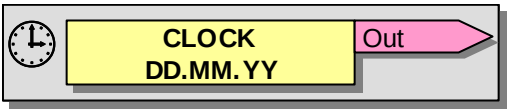


Datentyp: Out Text

Funktion: Diese Funktion liefert die aktuelle Uhrzeit als 8 stelligen Text mit dem 24h-Format HH:MM:SS.

Objekte:Uhr:Text:Datum

Symbol:

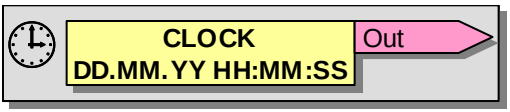


Datentyp: Out Text

Funktion: Diese Funktion liefert das aktuelle Datum als 8 stelligen Text mit dem Format TT.MM.JJ.

Objekte:Uhr:Text:Datum+Uhrzeit

Symbol:

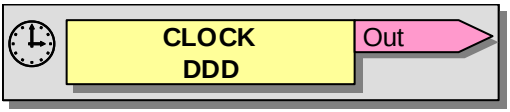


Datentyp: Out Text

Funktion: Diese Funktion liefert das aktuelle Datum und die aktuelle Uhrzeit als 17 stelligen Text mit dem Format TT.MM.JJ HH:MM:SS.

Objekte:Uhr:Text:Wochentag

Symbol:

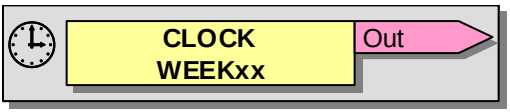


Datentyp: Out Text

Funktion: Diese Funktion liefert den aktuellen Wochentag als 3 stelligen Text. Die Wochentage werden nach der englischen schreibweise abgekürzt: MON,TUE,WED,THU,FRI,SAT,SUN.

Objekte:Uhr:Text:Woche

Symbol:

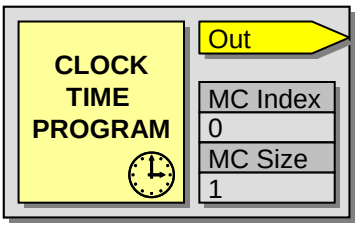


Datentyp: Out Text

Funktion: Diese Funktion liefert die aktuelle Kalenderwoche als 6 stelligen Text im Format WEEKxx.

Objekte:Uhr:Zeitprogramm

Symbol:



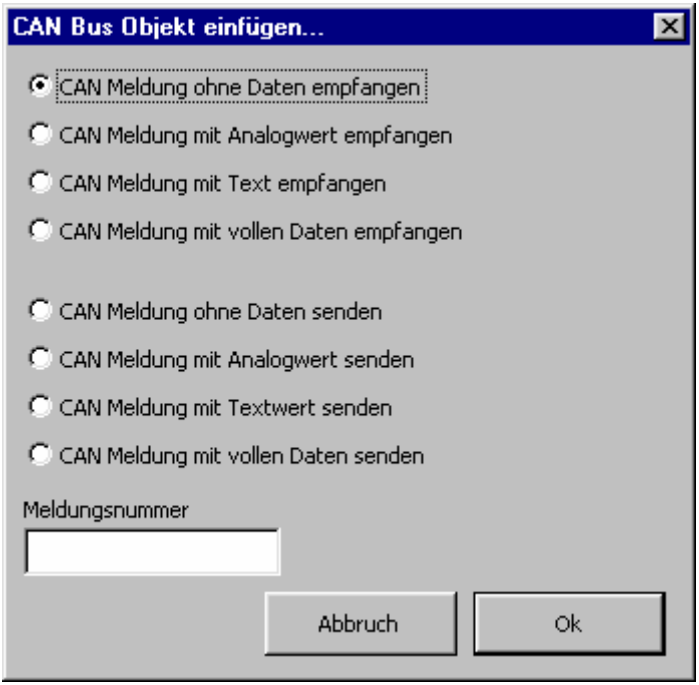
Datentyp: Out Bit

Funktion: Diese Funktion liefert einen Bitausgang, je nachdem ob das angegebene Zeitprogramm in der Memorycard gerade aktiv ist oder nicht.

CAN Objekte

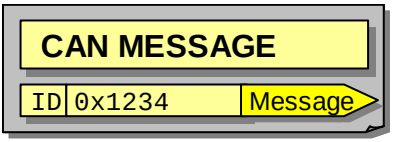
Dieser Abschnitt beschäftigt sich mit den CAN Funktionalitäten, die manche Geräte zur Verfügung stellen.

Um CAN-Objekte einzufügen, wählen Sie den Menüpunkt Objekte/CAN bus aus. Es erscheint folgende Auswahl:


CAN Bus Objekt einfügen...
CAN Meldung ohne Daten empfangen
CAN Meldung mit Analogwert empfangen
CAN Meldung mit Text empfangen
CAN Meldung mit vollen Daten empfangen
CAN Meldung ohne Daten senden
CAN Meldung mit Analogwert senden
CAN Meldung mit Textwert senden
CAN Meldung mit vollen Daten senden
Meldungsnummer
Abbruch
Ok

Objekte:CAN Meldung ohne Daten empfangen

Symbol:

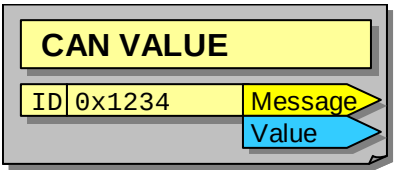


Datentyp: Message Bit

Funktion: Wenn am CAN Bus die Meldung 0x1234 empfangen wurde, wird der Ausgang Message für einen Zyklus aktiviert.

Objekte:CAN Meldung mit Analogwert empfangen

Symbol:



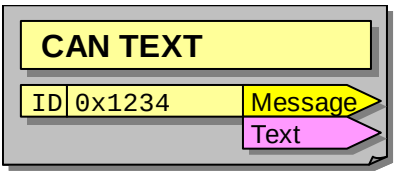
Datentyp: Message Bit

Value Analog

Funktion: Wenn am CAN Bus die Meldung 0x1234 empfangen wurde, wird der Ausgang Message für einen Zyklus aktiviert und der empfangene Analogwert wird am Ausgang Value zur Verfügung gestellt.

Objekte:CAN Meldung mit Text empfangen

Symbol:



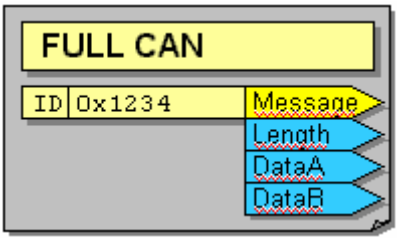
Datentyp: Message Bit

Text Text

Funktion: Wenn am CAN Bus die Meldung 0x1234 empfangen wurde, wird der Ausgang Message für einen Zyklus aktiviert und der empfangene Textwert wird am Ausgang Value zur Verfügung gestellt.

Objekte:CAN Meldung mit vollen Daten empfangen

Symbol:

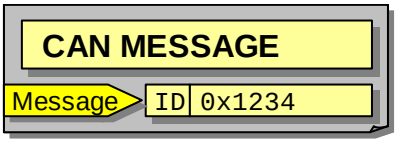


Datentyp: Message Bit
Length Analog
DataA Analog
DataB Analog

Funktion:

Objekte:CAN Meldung ohne Daten senden

Symbol:

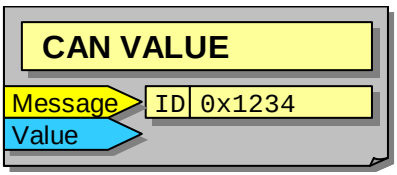


Datentyp: Message Bit

Funktion: Wenn am Eingang Message eine steigende Flanke erkannt wird, sendet das CAN Modul die Meldung 0x1234 über den CAN Bus.

Objekte:CAN Meldung mit Analogwert senden

Symbol:



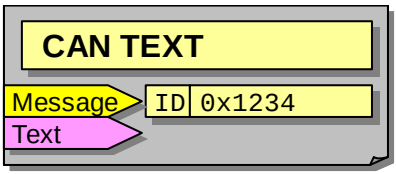
Datentyp: Message Bit

Value Analog

Funktion: Wenn am Eingang Message eine steigende Flanke erkannt wird, sendet das CAN Modul die Meldung 0x1234 über den CAN Bus. Im Datenbereich des CAN Frames wird der Analogwert Value mit gesendet.

Objekte:CAN Meldung mit Textwert senden

Symbol:



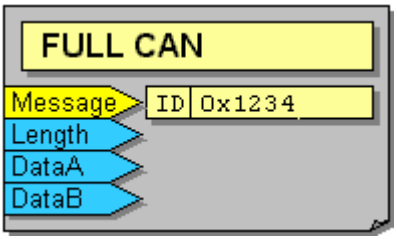
Datentyp: Message Bit

Text Text

Funktion: Wenn am Eingang Message eine steigende Flanke erkannt wird, sendet das CAN Modul die Meldung 0x1234 über den CAN Bus. Im Datenbereich des CAN Frames wird der Textwert Value mit gesendet.

Objekte:CAN Meldung vollen Daten senden

Symbol:



Datentyp: Message Bit

Length Analog

DataA Analog

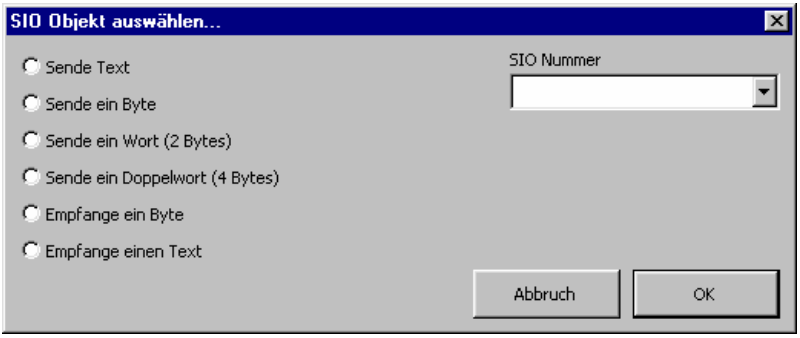
DataB Analog

Funktion: Wenn am Eingang Message eine steigende Flanke erkannt wird, sendet das CAN Modul die Meldung 0x1234 über den

CAN Bus. Im Datenbereich des CAN Frames wird der Analogwert Value mit gesendet.

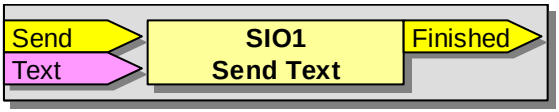
SIO Funktionen

Einige Module besitzen freie serielle Schnittstellen. Diese können mit den folgenden Funktionen angesprochen werden. Wählen Sie dazu den Menüpunkt Objekte/Serielles Interface an. Es erscheint folgende Auswahl:



Objekte:SIO:Sende Text

Symbol:

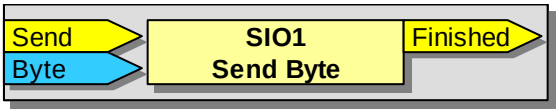


Datentyp:	Send	Bit
	Text	Text
	Finished	Bit

Funktion: Diese Funktion sendet bei einer steigenden Flanke am Eingang Send die Zeichenkette, die am Eingang Text anliegt. Wenn der Sendevorgang abgeschlossen ist, wird der Ausgang Finished aktiv.

Objekte:SIO:Sende ein Byte

Symbol:



Datentyp: Send Bit

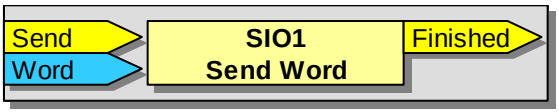
Byte Analog

Finished Bit

Funktion: Diese Funktion sendet bei einer steigenden Flanke am Eingang Send die untersten 8 Bits des Analogwertes am Eingang Byte als ein Zeichen. Wenn der Sendevorgang abgeschlossen ist, wird der Ausgang Finished aktiv.

Objekte:SIO:Send ein Wort(2 Bytes)

Symbol:



Datentyp: Send Bit

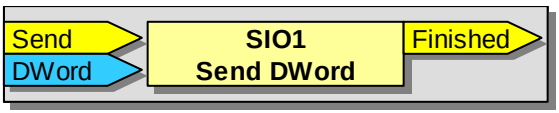
Word Analog

Finished Bit

Funktion: Diese Funktion sendet bei einer steigenden Flanke am Eingang Send die untersten 16 Bits des Analogwertes am Eingang Word als zwei aufeinander folgende Zeichen. Zuerst werden die untersten 8 Bits als ein Zeichen versendet. Danach werden die Bits 8 bis 15 als ein zweites Zeichen versendet. Wenn der Sendevorgang abgeschlossen ist, wird der Ausgang Finished aktiv.

Objekte:SIO:Sende ein Doppelwort(4 Bytes)

Symbol:



Datentyp: Send Bit

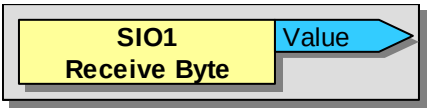
DWord Analog

Finished Bit

Funktion: Diese Funktion sendet bei einer steigenden Flanke am Eingang Send alle 32 Bits des Analogwertes am Eingang Word als vier aufeinander folgende Zeichen. Zuerst werden die untersten 8 Bits als ein Zeichen versendet. Danach werden die Bits 8 bis 15 als ein zweites Zeichen versendet. Dann folgen die Bits 16 bis 24 als drittes Zeichen. Zuletzt werden die obersten 8 Bits als viertes Zeichen versendet. Wenn der Sendevorgang abgeschlossen ist, wird der Ausgang Finished aktiv.

Objekte:SIO:Empfange ein Byte

Symbol:

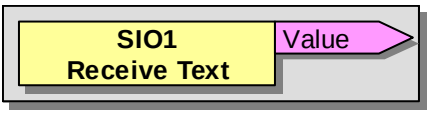


Datentyp: Value Analog

Funktion: Diese Funktion prüft, ob ein Zeichen auf der seriellen Schnittstelle empfangen wurde. Ist dies der Fall, so retourniert Value in den untersten 8 Bits den Code des empfangenen Zeichens. Wurde kein Zeichen empfangen wird der Analogwert 9999.000 zurückgegeben.

Objekte:SIO :Empfange einen Text

Symbol:

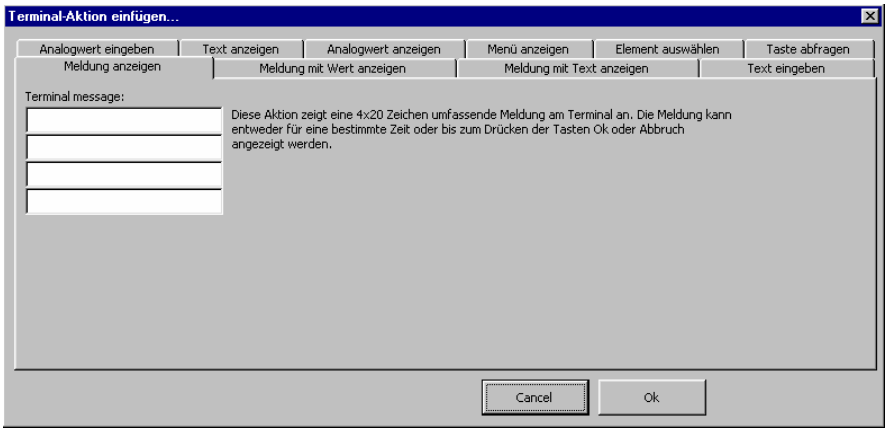


Datentyp: Value Text

Funktion: Diese Funktion prüft, ob ein Zeichen auf der seriellen Schnittstelle empfangen wurde. Ist dies der Fall, so wird ein 1 Zeichen langer Text mit dem empfangenen Zeichen zurückgegeben. Wurde kein Zeichen empfangen wird ein leerer Text zurückgegeben.

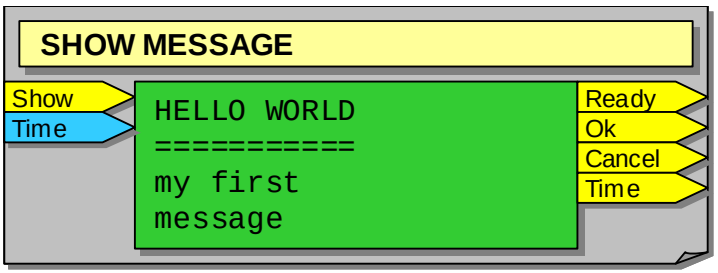
Terminalfunktionen

RESI'S GRAPH unterstützt zwei verschiedene Terminals. Das RESI-TERM4 und das RESI-TERM5. Beide Terminals werden ident angesprochen. Wählen Sie den Menüpunkt Objekte/Terminal an. Es erscheint folgende Auswahlmaske:


Terminal-Aktion einfügen...
Analogwert eingeben
Text anzeigen
Analogwert anzeigen
Menü anzeigen
Element auswählen
Taste abfragen
Meldung anzeigen
Meldung mit Wert anzeigen
Meldung mit Text anzeigen
Text eingeben
Terminal message:
Diese Aktion zeigt eine 4x20 Zeichen umfassende Meldung am Terminal an. Die Meldung kann entweder für eine bestimmte Zeit oder bis zum Drücken der Tasten Ok oder Abbruch angezeigt werden.
Cancel
Ok

Objekte:Terminal:Meldung anzeigen

Symbol:



Eingabe: Hier wird der Text erfasst, der am Terminal ausgegeben werden soll. Der Text darf exakt 4x20 Zeichen lang sein!

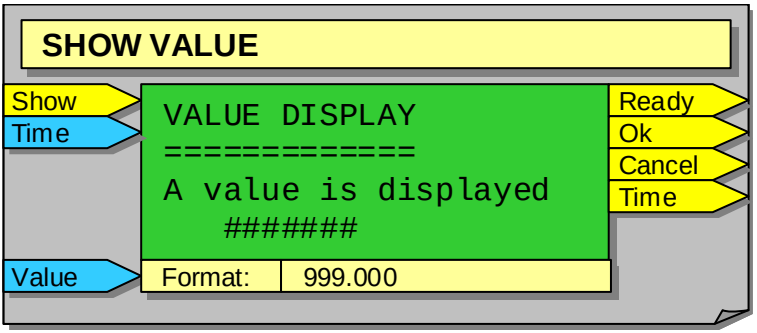
Datentyp:

Show	Bit
Time	Analog
Ready	Bit
Ok	Bit
Cancel	Bit
Time	Bit

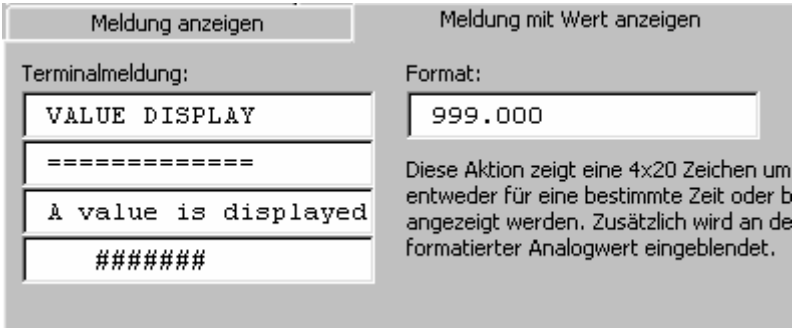
Funktion: Mit dieser Funktion wird der hinterlegte Text am Terminal angezeigt, wenn am Eingang Show eine steigende Flanke anliegt. Zusätzlich kann am Eingang Time eine Anzeigezeit in Sekunden vorgegeben werden. Läuft diese Zeit ab, wird der Ausgang Time aktiv. Bleibt dieser Eingang unbeschaltet, wird die Zeitfunktion ignoriert. Der Ausgang Ready wird dann aktiv, wenn der komplette Bildschirmaufbau abgeschlossen ist. Drückt der Bediener die OK Taste, so wird der Ausgang OK aktiv. Drückt der Bediener die Cancel Taste, so wird der Ausgang Cancel aktiv.

Objekte:Terminal:Meldung mit Wert anzeigen

Symbol:



Eingabe: Hier wird der Text erfasst, der am Terminal ausgegeben werden soll. Der Text darf exakt 4x20 Zeichen lang sein. Zusätzlich wird innerhalb des Textes durch Eingabe von # Zeichen der Platz markiert, wo die Zahl erscheinen soll. Im Feld Value format wird das Zahlenformat für die Anzeige eingegeben.



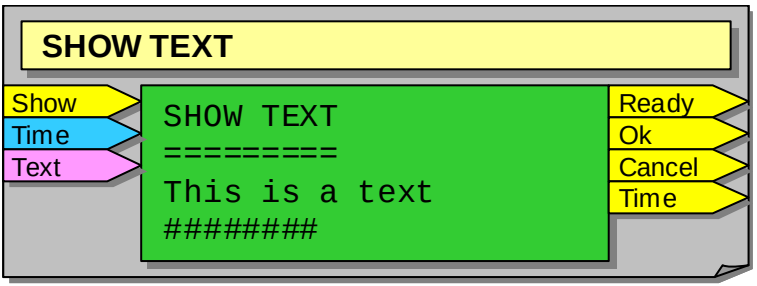
Datentyp:

Show	Bit
Time	Analog
Value	Analog
Ready	Bit
Ok	Bit
Cancel	Bit
Time	Bit

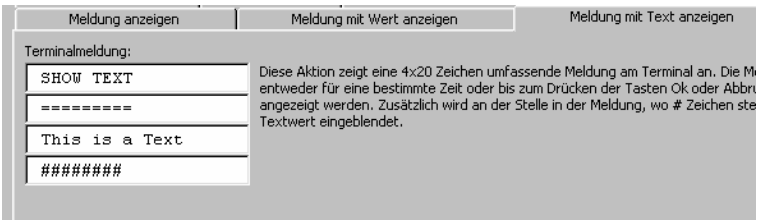
Funktion: Mit dieser Funktion wird der hinterlegte Text am Terminal angezeigt, wenn am Eingang Show eine steigende Flanke anliegt. Zusätzlich kann am Eingang Time eine Anzeigezeit in Sekunden vorgegeben werden. Läuft diese Zeit ab, wird der Ausgang Time aktiv. Bleibt dieser Eingang unbeschaltet, wird die Zeitfunktion ignoriert. Der Ausgang Ready wird dann aktiv, wenn der komplette Bildschirmaufbau abgeschlossen ist. Drückt der Bediener die OK Taste, so wird der Ausgang OK aktiv. Drückt der Bediener die Cancel Taste, so wird der Ausgang Cancel aktiv. Am Eingang Value liegt der Wert an, der mit dem Format anstelle der #-Zeichen angezeigt wird.

Objekte:Terminal:Meldung mit Text anzeigen

Symbol:



Eingabe: Hier wird der Text erfasst, der am Terminal ausgegeben werden soll. Der Text darf exakt 4x20 Zeichen lang sein. Zusätzlich wird innerhalb des Textes durch Eingabe von # Zeichen der Platz markiert, wo der Text erscheinen soll.


Meldung anzeigen
Meldung mit Wert anzeigen
Meldung mit Text anzeigen
Terminalmeldung:
SHOW TEXT
=========
This is a Text
########
Diese Aktion zeigt eine 4x20 Zeichen umfassende Meldung am Terminal an. Die M
entweder für eine bestimmte Zeit oder bis zum Drücken der Tasten Ok oder Abbr
angezeigt werden. Zusätzlich wird an der Stelle in der Meldung, wo # Zeichen ste
Textwert eingeblendet.

Datentyp:

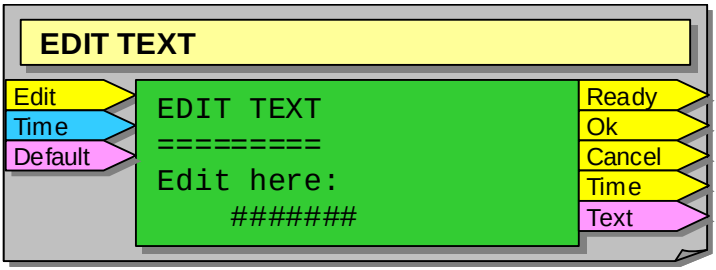
Show	Bit
Time	Analog
Text	Text

	Ready	Bit
	Ok	Bit
	Cancel	Bit
	Time	Bit

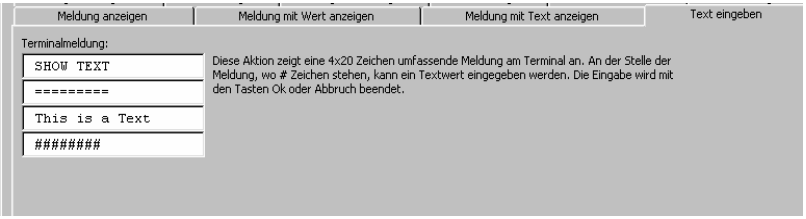
Funktion: Mit dieser Funktion wird der hinterlegte Text am Terminal angezeigt, wenn am Eingang Show eine steigende Flanke anliegt. Zusätzlich kann am Eingang Time eine Anzeigezeit in Sekunden vorgegeben werden. Läuft diese Zeit ab, wird der Ausgang Time aktiv. Bleibt dieser Eingang unbeschaltet, wird die Zeitfunktion ignoriert. Der Ausgang Ready wird dann aktiv, wenn der komplette Bildschirmaufbau abgeschlossen ist. Drückt der Bediener die OK Taste, so wird der Ausgang OK aktiv. Drückt der Bediener die Cancel Taste, so wird der Ausgang Cancel aktiv. Am Eingang Text liegt die Zeichenkette an, die anstelle der #-Zeichen angezeigt wird.

Objekte:Terminal:Text eingeben

Symbol:



Eingabe: Hier wird der Text erfasst, der am Terminal ausgegeben werden soll. Der Text darf exakt 4x20 Zeichen lang sein. Zusätzlich wird innerhalb des Textes durch Eingabe von # Zeichen der Platz markiert, wo die Texteingabe stattfinden kann.


Meldung anzeigen
Meldung mit Wert anzeigen
Meldung mit Text anzeigen
Text eingeben
Terminalmeldung:
SHOW TEXT
=========
This is a Text
########
Diese Aktion zeigt eine 4x20 Zeichen umfassende Meldung am Terminal an. An der Stelle der Meldung, wo # Zeichen stehen, kann ein Textwert eingegeben werden. Die Eingabe wird mit den Tasten Ok oder Abbruch beendet.

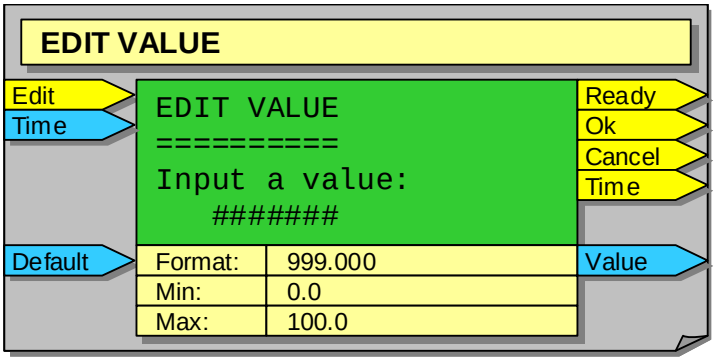
Datentyp:		
	Show	Bit
	Time	Analog
	Default	Text
	Text	Text
	Ready	Bit
	Ok	Bit
	Cancel	Bit
	Time	Bit

Funktion: Mit dieser Funktion wird der hinterlegte Text am Terminal angezeigt, wenn am Eingang Show eine steigende Flanke anliegt. Zusätzlich kann am Eingang Time eine Anzeigezeit in Sekunden vorgegeben werden. Läuft diese Zeit ab, wird der Ausgang Time aktiv. Bleibt dieser Eingang unbeschaltet, wird die Zeitfunktion ignoriert. Der Ausgang Ready wird dann aktiv, wenn der komplette Bildschirmaufbau abgeschlossen ist. Drückt der Bediener die OK Taste, so wird der Ausgang OK aktiv. Drückt der Bediener die Cancel Taste, so wird der Ausgang Cancel aktiv.

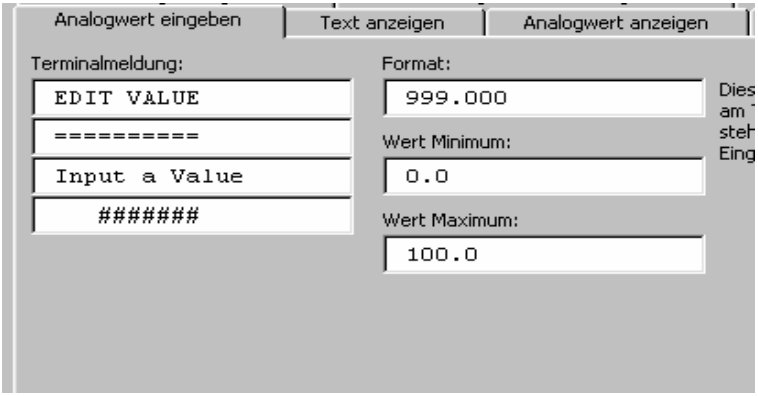
Der Eingang Default definiert die Zeichenkette, die beim Start der Eingabe angezeigt und durch den Benutzer editiert werden kann. Drückt der Benutzer die OK-Taste, so liegt am Ausgang Text der neu eingegebene Text an. Drückt der Benutzer die Taste Cancel, so wird die aktuelle Eingabe verworfen und der Text Default wird auch am Ausgang Text geliefert.

Objekte:Terminal:Analogwert eingeben

Symbol:



Eingabe: Hier wird der Text erfasst, der am Terminal ausgegeben werden soll. Der Text darf exakt 4x20 Zeichen lang sein. Zusätzlich wird innerhalb des Textes durch Eingabe von # Zeichen der Platz markiert, wo die Zahleneingabe stattfinden kann. Im Feld Value format wird das Zahlenformat für die Anzeige des Wertes angegeben. Die beiden Felder Value minimum und Value maximum definieren einen Zahlenbereich, in dem eine gültige Eingabe liegen muss.



Datentyp:

Show	Bit
Time	Analog
Default	Analog
Value	Analog

Ready Bit

Ok Bit

Cancel Bit

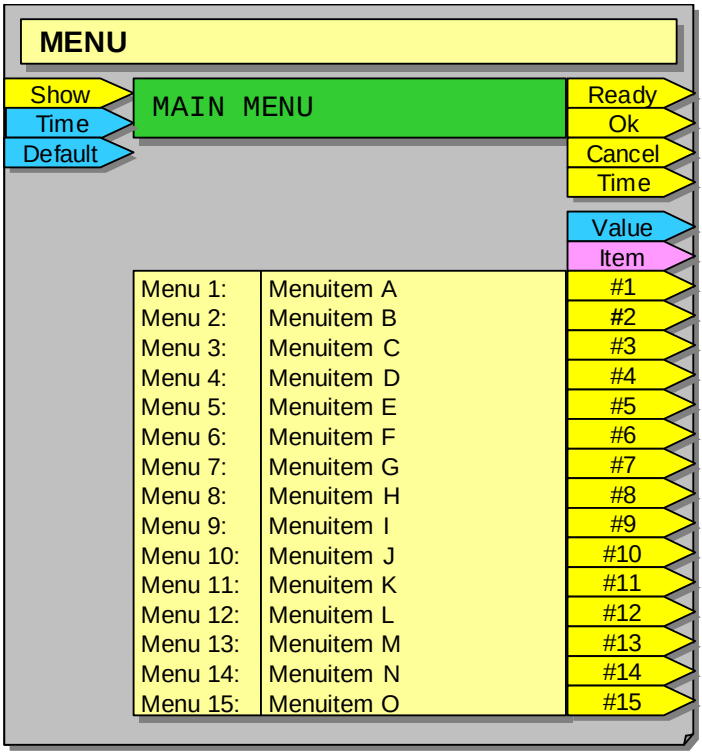
Time Bit

Funktion: Mit dieser Funktion wird der hinterlegte Text am Terminal angezeigt, wenn am Eingang Show eine steigende Flanke anliegt. Zusätzlich kann am Eingang Time eine Anzeigezeit in Sekunden vorgegeben werden. Läuft diese Zeit ab, wird der Ausgang Time aktiv. Bleibt dieser Eingang unbeschaltet, wird die Zeitfunktion ignoriert. Der Ausgang Ready wird dann aktiv, wenn der komplette Bildschirmaufbau abgeschlossen ist. Drückt der Bediener die OK Taste, so wird der Ausgang OK aktiv. Drückt der Bediener die Cancel Taste, so wird der Ausgang Cancel aktiv.

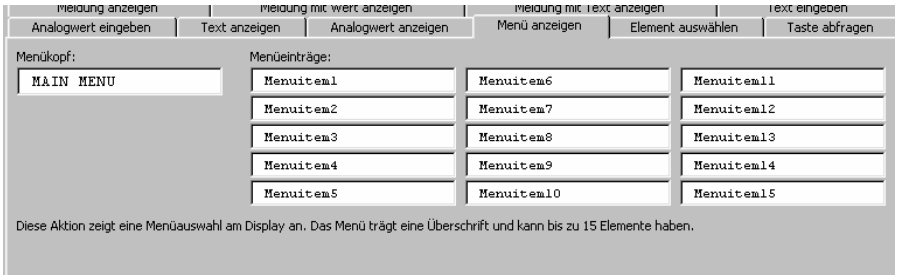
Der Eingang Default definiert den Zahlenwert, der beim Start der Zahleneingabe dargestellt wird. Drückt der Bediener nach dem Eingeben der Zahl die Taste OK, so wird die Eingabe konvertiert und der Analogwert wird auf die beiden Grenzen Value minimum und Value maximum untersucht. Liegt der eingegebene Wert im Bereich so wird der Ausgang OK aktiviert und an Ausgang Value wird der eingegebene Wert geliefert. Ist die eingegebene Zahl außerhalb des Bereichs, kann die Eingabe nicht mit OK fortgesetzt werden. Drückt man die CANCEL Taste, so wird der Wert Default als Ergebnis der Eingabe geliefert.

Objekte:Terminal:Menü anzeigen

Symbol:



Eingabe: Im Feld Menu header kann die Titelzeile des Auswahlmenüs eingegeben werden. Unter der Rubrik Menu items können bis zu 15 Menüeinträge definiert werden, die dann am Display ausgewählt werden.



Datentyp: Show Bit

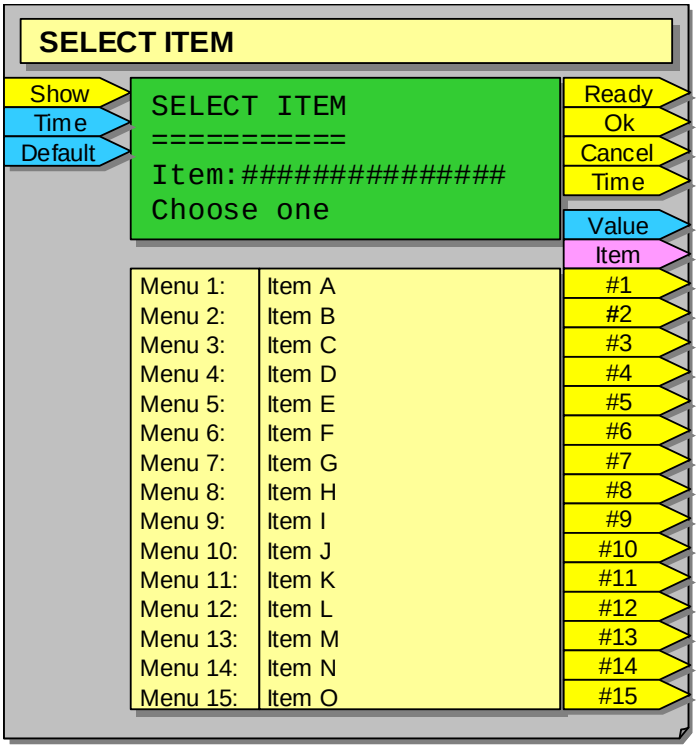
Time	Analog
Default	Analog
Value	Analog
Item	Text
Ready	Bit
Ok	Bit
Cancel	Bit
Time	Bit
#1-#15	Bit

Funktion: Mit dieser Funktion wird der hinterlegte Text am Terminal angezeigt, wenn am Eingang Show eine steigende Flanke anliegt. Zusätzlich kann am Eingang Time eine Anzeigezeit in Sekunden vorgegeben werden. Läuft diese Zeit ab, wird der Ausgang Time aktiv. Bleibt dieser Eingang unbeschaltet, wird die Zeitfunktion ignoriert. Der Ausgang Ready wird dann aktiv, wenn der komplette Bildschirmaufbau abgeschlossen ist. Drückt der Bediener die OK Taste, so wird der Ausgang OK aktiv. Drückt der Bediener die Cancel Taste, so wird der Ausgang Cancel aktiv.

Es wird ein Menü aufgebaut. Der selektiert Eintrag entspricht dem Eingang Default. Ist dieser Eingang unbeschaltet, so wird immer der erste Menüeintrag selektiert. Hat man einen Menüeintrag ausgewählt und drückt man die OK Taste, so wird an Ausgang Value die Nummer des gewählten Menüeintrages ausgegeben. Zusätzlich wird am Ausgang Item der Menüeintragstext ausgegeben und der entsprechende Ausgang #1 bis #15 wird ebenfalls aktiviert. Beim Drücken von Cancel wird nur der Ausgang Cancel aktiviert.

Objekte:Terminal:Element auswählen

Symbol:



Eingabe: Im Feld Terminal message wird der Text für das Display eingetragen. An der Stelle, wo die Auswahl der Einträge erscheinen soll, werden #-Zeichen in den Text eingetragen. Unter der Rubrik Items to select können bis zu 15 Einträge definiert werden, die dann am Display ausgewählt werden.

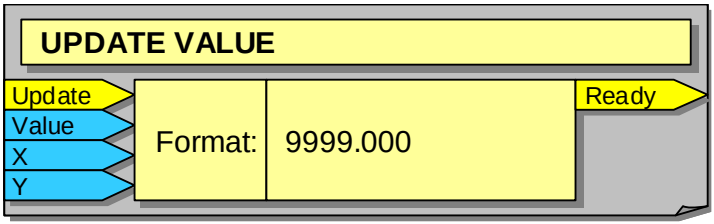
Datentyp:		
	Show	Bit
	Time	Analog
	Default	Analog
	Value	Analog
	Item	Text
	Ready	Bit
	Ok	Bit
	Cancel	Bit
	Time	Bit
	#1-#15	Bit

Funktion: Mit dieser Funktion wird der hinterlegte Text am Terminal angezeigt, wenn am Eingang Show eine steigende Flanke anliegt. Zusätzlich kann am Eingang Time eine Anzeigezeit in Sekunden vorgegeben werden. Läuft diese Zeit ab, wird der Ausgang Time aktiv. Bleibt dieser Eingang unbeschaltet, wird die Zeitfunktion ignoriert. Der Ausgang Ready wird dann aktiv, wenn der komplette Bildschirmaufbau abgeschlossen ist. Drückt der Bediener die OK Taste, so wird der Ausgang OK aktiv. Drückt der Bediener die Cancel Taste, so wird der Ausgang Cancel aktiv.

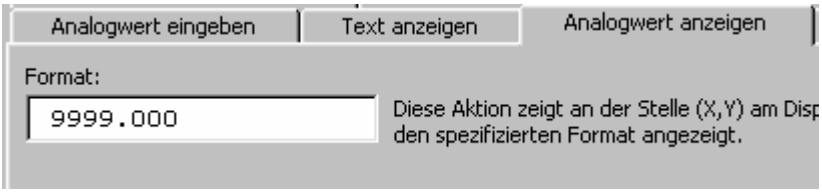
Es wird die eingegebene Meldung aufgebaut. Der dargestellte Eintrag entspricht dem Eingang Default. Ist dieser Eingang unbeschaltet, so wird immer der erste Eintrag dargestellt. Hat man einen Eintrag ausgewählt und drückt man die OK Taste, so wird an Ausgang Value die Nummer des gewählten Eintrages ausgegeben. Zusätzlich wird am Ausgang Item der Eintragstext ausgegeben und der entsprechende Ausgang #1 bis #15 wird ebenfalls aktiviert. Beim Drücken von Cancel wird nur der Ausgang Cancel aktiviert.

Objekte:Terminal:Analogwert anzeigen

Symbol:



Eingabe: Im Feld Value format wird das Format der Analogzahl eingegeben.



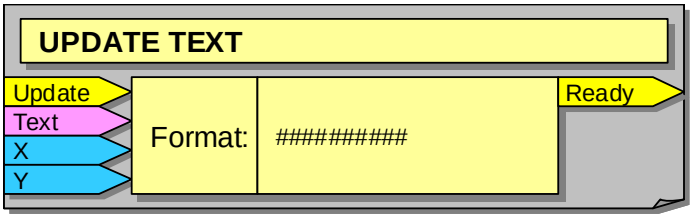
Datentyp:

Update	Bit
Value	Analog
X	Analog
Y	Analog
Ready	Bit

Funktion: Der aktuelle Wert des Einganges Value wird mit dem spezifizierten Format formatiert und in den aktuellen Terminalinhalt ab der Position X und Y eingeblendet. Die Koordinaten werden mit (0,0) beginnend gezählt. Diese Aktion wird bei jeder steigenden Flanke des Einganges Update ausgeführt. Der Ausgang Ready wird sofort nach der Darstellung der Zahl aktiviert.

Objekte:Terminal:Text anzeigen

Symbol:



Eingabe: Im Feld Text format wird die Länge des auszugebenden textes mit #-Zeichen festgelegt.

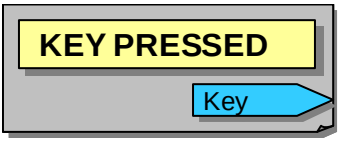
Datentyp:

Update	Bit
Text	Text
X	Analog
Y	Analog
Ready	Bit

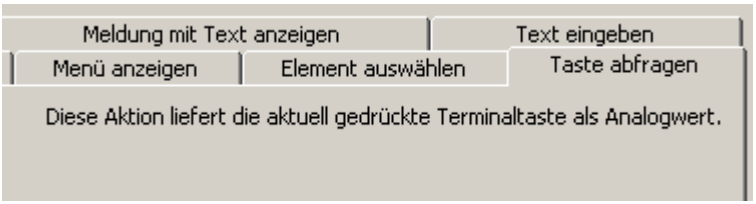
Funktion: Der aktuelle Wert des Einganges Text wird mit der spezifizierten Länge in den aktuellen Terminalinhalt ab der Position X und Y eingeblendet. Die Koordinaten werden mit (0,0) beginnend gezählt. Diese Aktion wird bei jeder steigenden Flanke des Einganges Update ausgeführt. Der Ausgang Ready wird sofort nach der Darstellung der Zahl aktiviert.

Objekte:Terminal:Taste abfragen

Symbol:



Eingabe: Dieser Befehl benötigt keine Parameter.



Datentyp: Key Analog

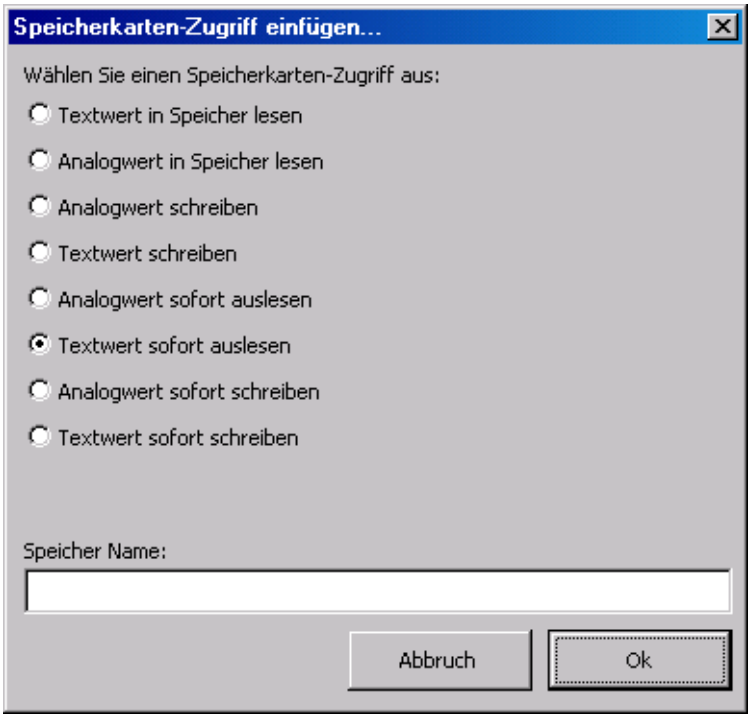
Funktion: Der aktuelle Wert der Tasten wird am Ausgang Key ausgegeben. Wurde keine Taste gedrückt, so wird der Wert 0 zurückgeliefert. Sonst wird folgender Code retourniert (Ganze Zahl zwischen 1 und 9):

2.000	5.000	9.000
3.000	7.000	4.000
8.000	6.000	1.000

Speicherkarte

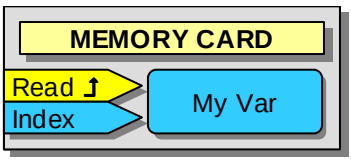
In diesem Abschnitt werden alle Funktionen behandelt, die das Speichern von Werten auf remanente Speicher wie der Speicherkarte ermöglichen. Die Speicherkarte kann 4096 Analogwerte und 512 Texte speichern.

Wählen Sie dazu den Menüpunkt Objekte/Speicherkarte Card aus. Es erscheint folgende Auswahl:



Objekte:Speicherkarte:Analogwert in Speicher lesen

Symbol:



Datentyp: Read Bit

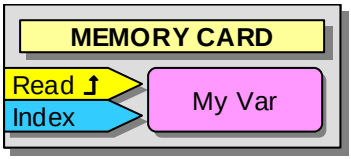
Index Analog

Funktion: Wird am Eingang Read eine steigende Flanke erkannt, wird aus der Speicherkarte der in der Position Index gespeicherte Analogwert ausgelesen und in die Variable MyVar abgespeichert.

HINWEIS: Werden die Adressen 100000 bis 100002 verwendet, so werden die gespeicherten Analogwerte aus der Echtzeituhr genommen. Wird ein Index im Bereich von 0 bis 4095 verwendet, so wird der entsprechende Wert aus der MemoryCard gelesen. Das Lesen eines Wertes aus der MemoryCard benötigt keine interne Verzögerung und kann somit beliebig oft in einem Programm benutzt werden.

Objekte:MemoryCard:Textwert in Speicher lesen

Symbol:



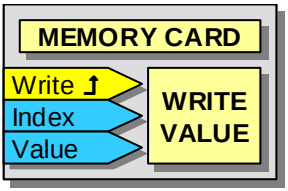
Datentyp: Read Bit

Index Analog

Funktion: Wird am Eingang Read eine steigende Flanke erkannt, wird aus der MemoryCard der in der Position Index gespeicherte Textwert ausgelesen und in die Textvariable MyVar abgespeichert. Wird ein Index im Bereich von 0 bis 511 verwendet, so wird der entsprechende Text aus der MemoryCard gelesen. Das Lesen eines Wertes aus der MemoryCard benötigt keine interne Verzögerung und kann somit beliebig oft in einem Programm benutzt werden.

Objekte:MemoryCard:Analogwert schreiben

Symbol:



Datentyp:

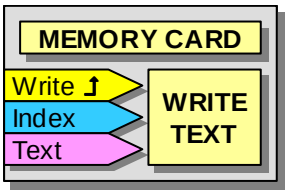
Write	Bit
Index	Analog
Value	Analog

Funktion: Wird am Eingang Write eine steigende Flanke erkannt, wird der am Eingang Value anliegende Analogwert in die MemoryCard an die Position Index geschrieben.

HINWEIS: Werden die Adressen 100000 bis 100002 verwendet, so werden die Analogwerte in die Echtzeituhr geschrieben. Wird ein Index im Bereich von 0 bis 4095 verwendet, so wird der entsprechende Wert in der MemoryCard beschrieben. Denken Sie auch daran, dass das Schreiben auf die MemoryCard ca.5 bis 10ms benötigt und nur ca. 100000 mal möglich ist! Danach ist die MemoryCard kaputt!

Objekte:MemoryCard:Textwert schreiben

Symbol:



Datentyp:

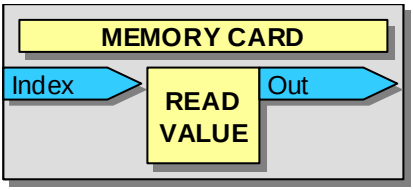
Write	Bit
Index	Analog
Text	Text

Funktion: Wird am Eingang Write eine steigende Flanke erkannt, wird der am Eingang Text liegende Textwert in die MemoryCard an die Position Index geschrieben.

HINWEIS: Wird ein Index im Bereich von 0 bis 511 verwendet, so wird der entsprechende Text in der MemoryCard beschrieben. Denken Sie auch daran, dass das Schreiben auf die MemoryCard ca.5 bis 10ms benötigt und nur ca. 100000 mal möglich ist! Danach ist die MemoryCard kaputt!

Objekte:MemoryCard:Analogwert lesen

Symbol:



Datentyp: Index Analog

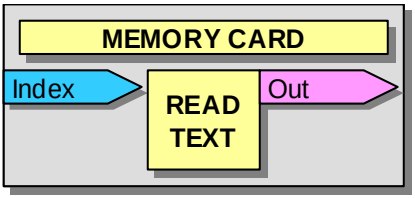
Out Analog

Funktion: Der aktuelle Wert am Analog Eingang Index wird am Analog Ausgang Out als Analog Variable gespeichert.

HINWEIS: Werden die Adressen 100000 bis 100002 verwendet, so werden die gespeicherten Analogwerte aus der Echtzeituhr genommen. Wird ein Index im Bereich von 0 bis 4095 verwendet, so wird der entsprechende Wert aus der MemoryCard gelesen. Das Lesen eines Wertes aus der MemoryCard benötigt keine interne Verzögerung und kann somit beliebig oft in einem Programm benutzt werden.

Objekte:MemoryCard:Texwert auslesen

Symbol:



Datentyp: Index Analog

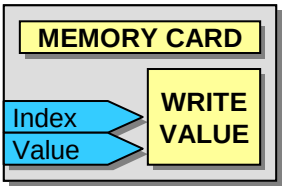
Out Text

Funktion: Der aktuelle Wert am Analog Eingang Index wird am Text Ausgang Out als Text Variable gespeichert.

Funktion: Wird am Eingang Read eine steigende Flanke erkannt, wird aus der MemoryCard der in der Position Index gespeicherte Textwert ausgelesen und in die Textvariable MyVar abgespeichert. Wird ein Index im Bereich von 0 bis 511 verwendet, so wird der entsprechende Text aus der MemoryCard gelesen. Das Lesen eines Wertes aus der MemoryCard benötigt keine interne Verzögerung und kann somit beliebig oft in einem Programm benutzt werden.

Objekte:MemoryCard:Analogwert sofort schreiben

Symbol:



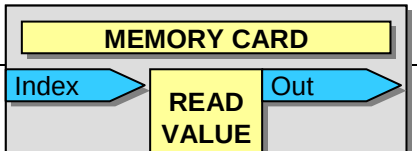
Funktion: Diese Funktion schreibt den Analogwert sofort in den Index 1001, das heißt man muss nicht auf eine steigende Flake des Digitaleinganges warten. Die Beschreibung des Index erfolgt sofort.

Datentypen:

Index	Analog
Value	Analog

Objekte:MemoryCard:Analogwert sofort lesen

Symbol:

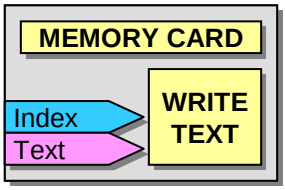


Funktion: Diese Funktion schreibt den Analogwert der, im Speicher 1001 liegt, sofort in den Analogspeicher „Out". Es muss nicht darauf gewartet werden, dass eine steigende Flanke auftritt!

Datentypen:		
	Index	Analog
	Out	Analog

Objekte:MemoryCard:Textwert sofort schreiben

Symbol:

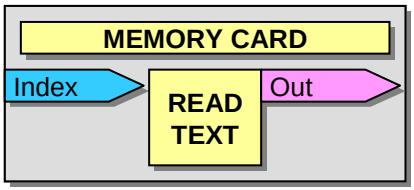


Funktion: Diese Funktion schreibt den Text sofort in den Index 1001, das heißt man muss nicht auf eine steigende Flake des Digitaleinganges warten. Die Beschreibung des Index erfolgt sofort.

Datentypen:		
	Index	Analog
	Text	Text

Objekte:MemoryCard:Textwert sofort lesen

Symbol:



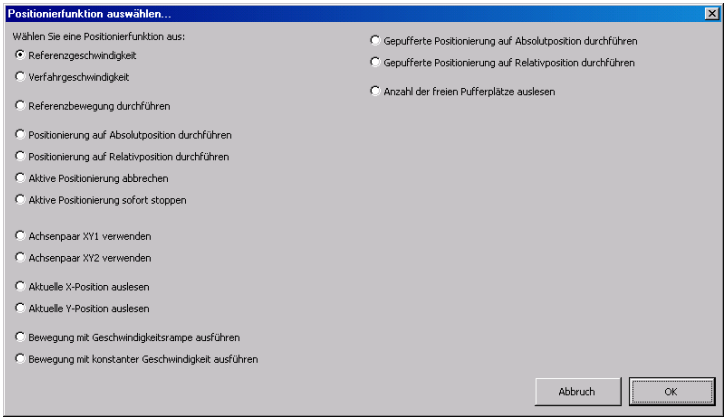
Funktion: Diese Funktion schreibt den Analogwert der, im Speicher 1001 liegt, sofort in den Analogspeicher „Out". Es muss nicht darauf gewartet werden, dass eine steigende Flanke auftritt!

Datentypen:	Index	Analog
	Out	Text

Positionierung

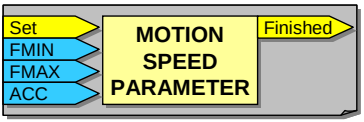
Die Funktion Positionierung ermöglicht Ihnen das genaue Ansteuern auf verschiedene Positionen. Um diese Funktion zu nutzen, wählen Sie die Funktion Positionierung unter dem Menü Objekte.

Es öffnet sich folgendes Fenster.



Objekte:Positionierung: Verfahrgeschwindigkeit

Symbol:



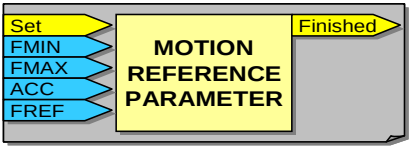
Funktion: Bei dieser Funktion werden die Geschwindigkeiten für die Bewegung eines Objektes eingestellt. Am Eingang FMIN wird die Geschw. eingestellt, mit welcher sich das Objekt zu bewegen beginnt. Am Eingang FMAX wird die höchste Geschw. angelegt, die das Objekt erreichen darf. Die Beschleunigung wird am Eingang ACC angelegt.

Datentypen:		
	Set	Binär
	FMIN	Analog

FMAX	Analog
ACC	Analog
Finished	Binär

Objekte:Positionierung: Referenzgeschwindigkeit

Symbol:



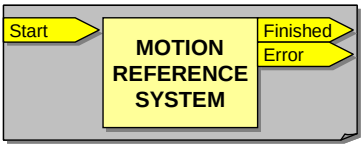
Funktion: Hier wird die Geschwindigkeiten eingestellt, die für die Referenzbewegung benötigt werden. Am Eingang FMIN wird die Geschwindigkeit angelegt, mit der sich das Objekt zu bewegen beginnt. Über den Eingang ACC wird die Beschleunigung eingegeben, mit der sich die Geschwindigkeit von FMIN auf FMAX ändert. Am Eingang FMAX wird jene Geschwindigkeit angelegt, mit der sich dann das Objekt in Richtung Sensor bewegt. Da das Objekt nicht sofort bremsen kann, fährt es über den Sensor. Nach dem stillstand des Objektes bewegt es sich mit einer sehr langsamen Geschwindigkeit in Richtung Sensor und bleibt stehen. Diese Geschwindigkeit wird als Referenzgeschwindigkeit bezeichnet und am Eingang FREF angelegt.

Datentypen:

Set	Binär
FMIN	Analog
FMAX	Analog
ACC	Analog
FREF	Analog
Finished	Binär

Objekte:Positionierung: Referenzbewegung durchführen

Symbol:

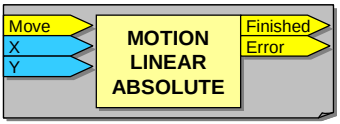


Funktion: Wird am Eingang Start eine steigende Flanke erkannt, bewegt sich das zu steuernde Objekt mit den Geschwindigkeiten, die bei den Parametern für die Referenzgeschwindigkeit angegeben sind, zurück auf seine Ausgangsposition.

Datentypen:		
	Start	Binär
	Finished	Binär
	Error	Binär

Objekte:Positionierung: Positionierung auf Absolutposition durchführen

Symbol:

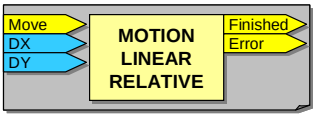


Funktion: Liegt am Eingang Move eine steigende Flanke an, wird das zu bewegende Objekt direkt auf die Koordinaten bewegt, welche an den Eingängen X und Y anliegen.

Datentypen:		
	Move	Binär
	X	Analog
	Y	Analog
	Finished	Binär
	Error	Binär

Objekte:Positionierung: Positionierung auf Relativposition durchführen

Symbol:

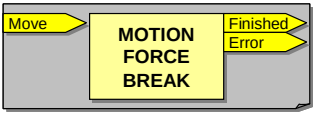


Funktion: Wird am Eingang Move eine steigende Flanke erkannt, wird das Objekt relativ auf die Koordinaten bewegt, welche an den Eingängen X und Y anliegen. Das bedeutet, der Wert der am Eingang anliegt, wir zu den Koordinaten der aktuelle Position addiert.

Datentypen:		
	Move	Binär
	DX	Analog
	DY	Analog
	Finished	Binär
	Error	Binär

Objekte:Positionierung: Aktive Positionierung abbrechen

Symbol:



Funktion: Tritt am Eingang Move eine steigende Flanke auf, wird die aktuelle Positionierung langsam angehalten. Das Objekt wird heruntergebremst, es ist kein ruckartiges stehen bleiben.

Datentypen:		
	Move	Binär
	Finished	Binär
	Error	Binär

Objekte:Positionierung: Aktive Positionierung sofort stoppen

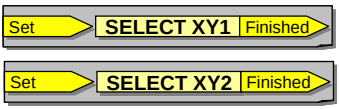
Symbol:

Funktion: Wird am Eingang Move eine steigende Flanke erkannt, wird das in Bewegung befindliche Objekt von seiner aktuellen Geschwindigkeit sofort abgebremst. Es ist eine ruckartiger Vorgang, wobei das Objekt jedoch sofort zum stehen kommt.

Datentypen:		
	Move	Binär
	Finished	Binär
	Error	Binär

Objekte:Positionierung: Achsenpaare verwenden

Symbol:



Funktion: Diese Funktion ist nötig, um zu definieren, welche Achsen verwendet werden sollen. In RESI's GRAPH gibt es zwei Auswahlmöglichkeiten: Achsenpaar XY1 und Achsenpaar XY2 Mit dem ersten Achsenpaar kann man die normalen x und y Koordinaten ansteuern. Mit dem zweiten Achsenpaar ist es möglich, auch die Z – Achse zu definieren.

Datentypen:		
	Set	Binär
	Finished	Binär

Objekte:Positionierung: Aktuelle X-Position

Symbol:

Funktion: Mit dieser Funktion liest man die Aktuelle Position an der X – Stelle aus.

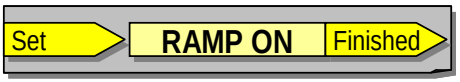
Objekte:Positionierung: Aktuelle Y-Position

Symbol:

Funktion: Mit dieser Funktion liest man die Aktuelle Position an der X – Stelle aus.

Objekte:Positionierung: Bewegung mit Geschwindigkeitsrampe ausführen

Symbol:



Funktion: Tritt am Eingang Set eine steigende Flanke auf, wird die RAMP ON aktiv. Die Rampe wird herunter gelassen und anschließend der Ausgang Finished aktiv.

Datentypen:

Set	Binär
Finished	Binär

Objekte:Positionierung: Bewegung mit konstanter Geschwindigkeit ausführen

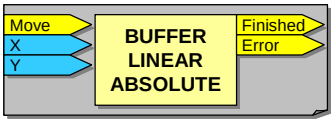
Symbol:

Funktion: Wird am Eingang Set eine steigende Flanke erkannt, wird die Funktion Ramp off aktiviert. Die Rampe wird geschlossen, wenn dies abgeschlossen ist, wird der Ausgang Finished aktiv.

Datentypen:		
	Set	Binär
	Finished	Binär

Objekte:Positionierung: Gepufferte Positionierung auf Asolutposition durchführen

Symbol:

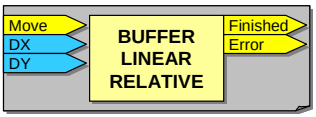


Funktion: Tritt am Eingang Move eine steigende Flanke auf, werden die Analogwerte für die X und Y Koordinaten eingelesen. Bei dieser Funktion können mehre Koordinaten eingelesen werde und diese in einem Puffer gespeichert werden, was dazu führ, dass das zu bewegende Objekt alle Koordinaten abfährt ab nie anhält.

Datentypen:		
	Move	Bit
	X	Analog
	Y	Analog
	Finished	Bit
	Error	Bit

Objekte:Positionierung: Gepufferte Positionierung auf Relativposition durchführen

Symbol:



Funktion: Tritt am Eingang Move eine steigende Flanke auf, werden die Analogwerte für die X und Y Koordinaten eingelesen und zur aktuelle Position addiert. Bei dieser Funktion können mehre Koordinaten eingelesen werde und in einem Puffer gespeichert werden, was dazu führ, dass das zu bewegende Objekt alle Koordinaten relativ abfährt ohne stehen zu bleiben.

Datentypen:		
	Move	Binär
	DX	Analog

DY	Analog
Finished	Binär
Error	Binär

Objekte:Positionierung: Anzahl der freien Pufferplätze auslesen.

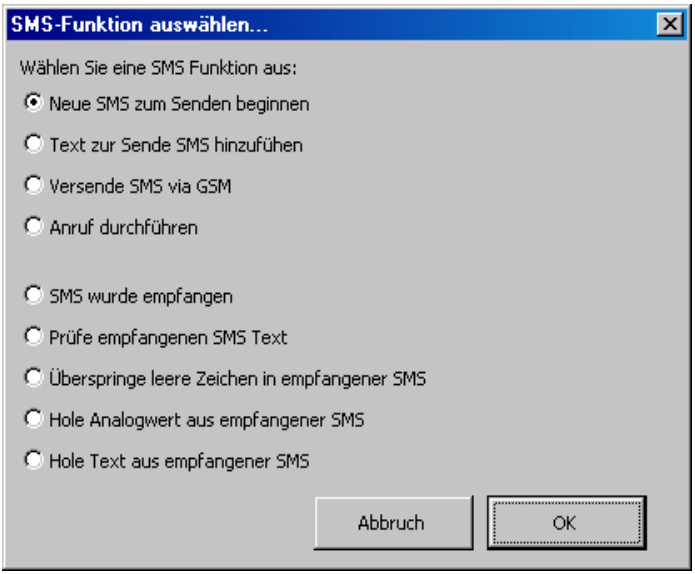
Symbol:

Funktion: Bei dieser Funktion wird die Anzahl der noch freien Pufferplätze ausgelesen und am Analogausgang X ausgegeben.

Datentypen: X Analog

SMS

Diese Funktion ermöglicht Ihnen das Senden und Empfangen von Nachrichten über ein GSM – Modem. Wählen Sie dazu das Menü SMS unter Objekte, es erschein folgendes Fenster:



Objekt:SMS:Neue SMS zum Senden beginnen

Symbol:

Funktion: Diese Funktion sendet bei einer steigenden Flanke am Eingang Start Steuerzeichen zum GSM – Modem, um auf Sendebetrieb zu gehen. Wenn der Sendevorgang abgeschlossen ist, wird der Ausgang Ready aktiv!

Datentypen:		
	Start	Bit
	Ready	Bit

Objekt:SMS:Text zur Sende SMS hinzufügen

Symbol:

Funktion: Tritt am Eingang Add eine steigende Flanke auf, wird der am Eingang Text anliegende Textwert eingelesen und der Nachricht hinzugefügt. Ist der Vorgang abgeschlossen wird der Ausgang Ready aktiv.

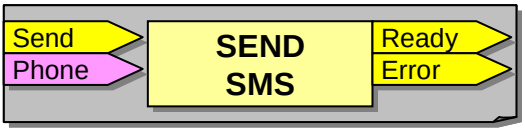
Hinweis: Es können max. 20 Zeichen auf ein Mal einer Nachricht hinzugeführt werden. Die maximale Größe einer Nachricht beträgt 60 Zeichen.

Datentypen:

Add	Bit
Text	Text
Ready	Analog

Objekt:SMS:Versende SMS via GSM

Symbol:



Funktion: Tritt am Eingang Send eine steigende Flanke auf, wird an die am Eingang Phone als Textwert anliegende Handynummer die vorher zusammengesetzte Nachricht versendet. Wurde die Nachricht erfolgreich versendet wird der Ausgang Ready aktiv. Sollte ein Fehler aufgetreten sein und die Nachricht nicht ordnungsgemäß an das GSM – Modem versendet sein, wir der Ausgang Error aktiv!

Datentypen:

Send	Bit
Phone	Text
Ready	Bit
Error	Bit

Objekt:SMS:Anruf durchführen

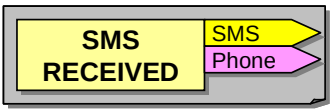
Symbol:

Funktion: Wird am Eingang Call eine 1 angelegt, wird an die am Eingang Phone als Textwert anliegende Handynummer ein Anruf so lange aufgebaut, bis am Eingang Call wieder eine 0 anliegt. Wenn der Anruf entgegengenommen wird, wird der Ausgang Ready aktiv. Sollte der Anruf nicht erfolgreich durchgeführt werde können, wird der Ausgang Error aktiv.

Datentypen:		
	Call	Bit
	Phone	Text
	Ready	Bit
	Error	Bit

Objekt:SMS:SMS wurde Empfangen

Symbol:

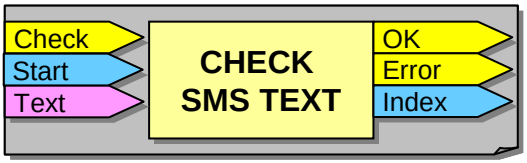


Funktion: Diese Funktion prüft ob eine Nachricht über das GSM-Modem empfangen wurde. Wenn dies der Fall ist, wird am Ausgang SMS eine steigende Flanke ausgegeben und am Ausgang Phone die Handynummer von der die Nachricht gesendet wurde.

Datentypen:		
	SMS	Bit
	Phone	Text

Objekt:SMS:Prüfe empfangenen SMS Text

Symbol:



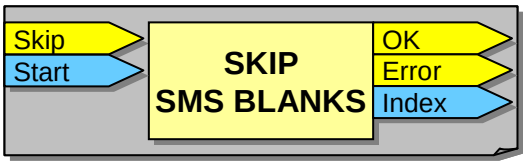
Funktion: Tritt am Eingang Check eine steigende Flanke auf, wird der Text der empfangenen Nachricht ab dem Startindex Start mit dem am Eingang Text anliegenden Textwert verglichen. Start wird mit 0 beginnend gezählt. Wenn der Text übereinstimmt wird der Ausgang OK aktiv. Stimmt der Text nicht überein, wird der Ausgang Error aktiv. Am Ausgang Index wird der analoge Wert der verglichenen Zeichen ausgegeben.

Datentypen:

Check	Bit
Start	Analog
Text	Text
OK	Bit
Error	Bit
Index	Analog

Objekt:SMS:Überspringe leere Zeichen in empfangener SMS

Symbol:



Funktion: Tritt am Eingang Skip eine steigende Flanke auf, wird der empfangene Text ab dem Startindex Start auf Leerzeichen überprüft. Start wird mit 0 beginnet gezählt. Wenn der Text vollständig überprüft wurde, wird der Ausgang OK aktiv. Sollte bei der Überprüfung ein Fehler aufgetreten sein, wird der Ausgang Error aktiv. Am Ausgang Index wird der analoge

Wert der erkannten Leerzeichen bis zum nächsten Textzeichen ausgegeben.

Datentypen:		
	Skip	Bit
	Start	Analog
	OK	Bit
	Error	Bit
	Index	Analog

Objekt:SMS:Hole Analogwert aus SMS

Symbol:

Funktion: Tritt am Eingang Get eine steigende Flanke auf, wird der Text der empfangenen Nachricht ab dem Startindex Start auf die Länge am Analogeingang Lenght als Analogwert eingelesen. Start wird mit 0 beginnet gezählt. Wenn der Analogwert am Ausgang Value ausgegeben wurde, wird der Ausgang OK aktiviert. Konnte der Analogwert nicht ausgegeben werde, wird der Ausgang Error aktiv. Am Ausgang Index wird der analoge Wert der erkannten Zeichen ausgegeben.

Datentypen:		
	Get	Bit
	Star	Analog
	Lenght	Analog
	OK	Bit
	Error	Bit
	Index	Analog
	Value	Analog

Objekt:SMS:Hole Text aus SMS

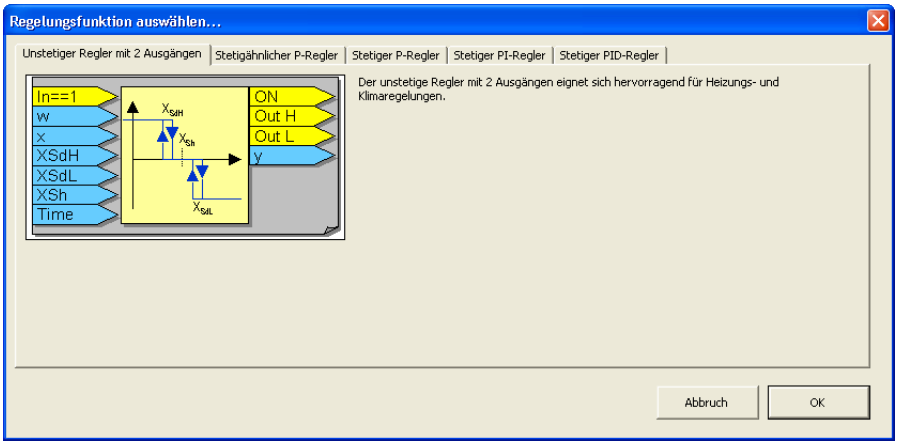
Symbol:

Funktion: Tritt am Eingang Get eine steigende Flanke auf, wird der Text der empfangenen Nachricht ab dem Startindex Start auf die Länge am Analogeingang Length als Textwert eingelesen. Start wird mit 0 beginnend gezählt. Wenn der Text am Ausgang Text ausgegeben wurde, wird der Ausgang OK aktiv. Konnte der Text nicht ausgegeben werden, wird der Ausgang Error aktiv. Am Ausgang Index wird der Analogwert der erkannten Zeichen ausgegeben.

Datentypen:

Get	Bit
Start	Analog
Lenghth	Analog
OK	Bit
Error	Bit
Index	Analog
Text	Text

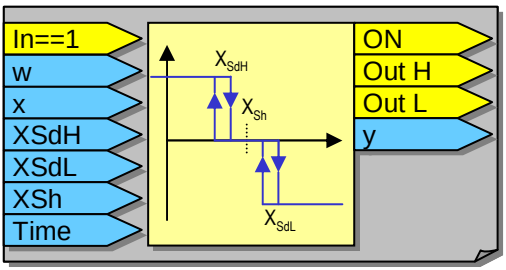
Regelungstechnik



Unter dieser Gruppe finden Sie alle regelungstechnischen Funktionen.

Objekt:Regelungstechnik: Unstetiger Hysteresen-Regler

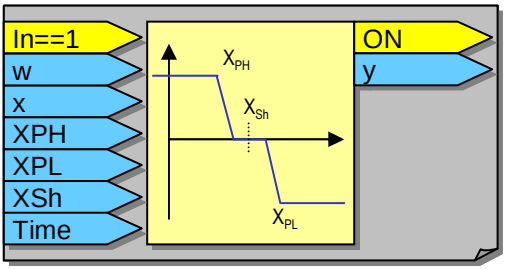
Symbol:



Funktion: Am Eingang x wird der zu regelnde Wert in Analogform angelegt. Der Wert auf den geregelt werden soll, wird am Eingang w in Analogform angelegt. Die Ausgangsgröße y kann nur drei Stati annehmen: -100..0..+100. Xsh legt die Hysterese um w fest. XSdH und XSdL legen nochmals zwei Hysterese für den Heizungs- und Kühlungsbereich fest. Zusätzlich werden die Ausgänge Out H bzw. Out L aktiviert, wenn x großer/kleiner w samt Hysterese ist. Am Analogeingang Time wird die Zeit angelegt, wie lange der Regelvorgang dauern soll.

Objekt:Regelungstechnik: Stetigähnlicher P-Regler

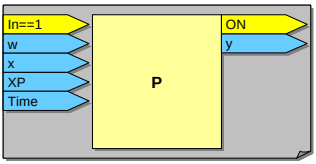
Symbol:



Funktion: Am Eingang x wird der zu regelnde Wert in Analogform angelegt. Der Wert auf den geregelt werden soll, wird am Eingang w in Analogform angelegt. Die Ausgangsgröße y wird nun analog der Grafik berechnet. Die beiden analogen Parameter Xph und Xpl stellen die Grenzen dar, ab wann 100% am Ausgang y generiert wird. Xsh legt die Hysterese rund um w fest. Am Analogeingang Time wird die Zeit angelegt, wie lange der Regelvorgang dauern soll.

Objekt:Regelungstechnik:Stetiger P-Regler

Symbol:

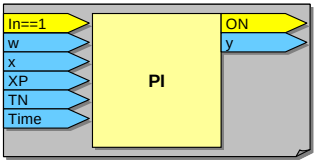


Funktion: Am Eingang x wird der zu regelnde Wert in Analogform angelegt. Der Wert auf den geregelt werden soll, wird am Eingang w in Analogform angelegt. Die Ausgangsgröße y kann sich nur innerhalb des Stellbereiches verändern. Eine Eingangsänderung, die eine Verstellung über den ganzen Stellbereich hervorruft, erstreckt sich über den Proportionalbereich, welchen man am Eingang X_p definieren kann.

Der Proportionalbereich X_p, wird in Prozenten des Normbereiches eingestellt. Am Analogeingang Time wird die Zeit angelegt, wie lange der Regelvorgang dauern soll.

Objekt:Regelungstechnik:Stetiger PI-Regler

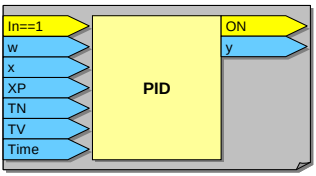
Symbol:



Funktion: Am Eingang x wird der zu regelnde Wert im Analogformat angelegt. Der Wert auf den zu regeln ist, wird im Analogformat an den Eingang w angelegt. Über den Analogeingang X_p kann man den Proportionalbereich verändern und über den Analogeingang T_N wird die Nachstellzeit angelegt. Dieser Regler besteht eigentlich aus einem I-Regler und einem P-Regler, die parallel geschalten sind. Daher ist die Zeit eine wichtige Kenngröße, die bei sprunghafter Einwirkung des I-Teiles benötigt wird, um die Wirkung des P-Teiles hervorzurufen. Diese Zeit wird als Nachstellzeit bezeichnet. Es ist zu beachten, dass die Nachstellzeit nur Einfluss auf den I-Teil des Reglers nimmt, wobei der Proportionalbereich sowohl auf den I-Teil und auch auf den P-Teil Einfluss nimmt.

Objekt:Regelungstechnik:Stetiger PID-Regler

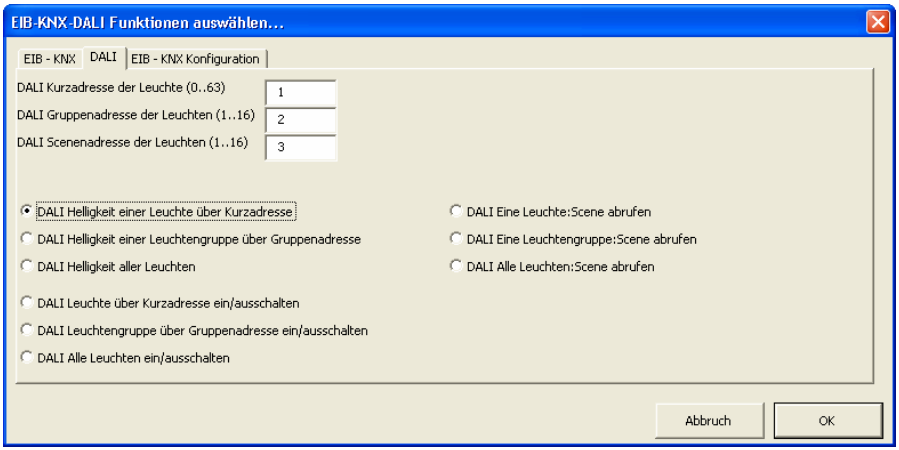
Symbol:



Funktion: Am Analogeingang x wird der Wert angelegt, der geregelt werden soll. Der Wert auf den geregelt werden soll, wird im Analogformat am Eingang w angelegt. Der Proportionalbereich wird über den Eingang X_P angelegt. Am Eingang T_N wird die Nachstellzeit angelegt und am Eingang T_V die Vorhaltezeit. Im

Grunde ist eine PID-Regler eine Parallelschaltung von einem P-Regler, einem I-Regler und einem D-Regler.

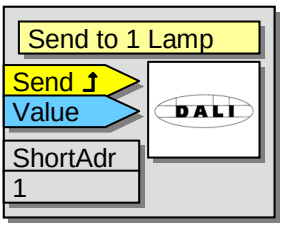
DALI



Unter dieser Funktionsgruppe sind alle DALI Steuerbefehle zusammengefasst.

Objekt:DALI:Helligkeit einer Leuchte über Kurzadresse

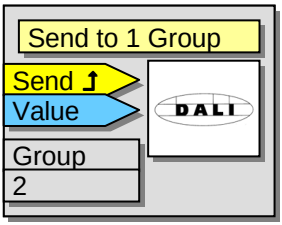
Symbol:



Funktion: Bei steigender Flanke am Eingang Send wird der Helligkeitswert Value in Prozent an die DALI Leuchte mit der Kurzadresse ShortAdr (zB:1) gesendet. Als Kurzadresse ist 0..63 zulässig.

Objekt:DALI:Helligkeit einer Leuchte über Gruppe

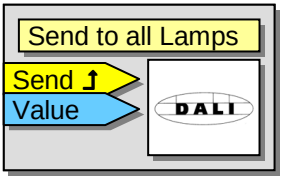
Symbol:



Funktion: Bei steigender Flanke am Eingang Send wird der Helligkeitswert Value in Prozent an alle DALI Leuchten mit der Gruppe Group (zB:2) gesendet. Als Gruppen ist 0..15 zulässig.

Objekt:DALI:Helligkeit aller Leuchten

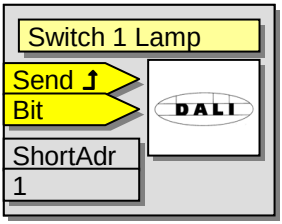
Symbol:



Funktion: Bei steigender Flanke am Eingang Send wird der Helligkeitswert Value in Prozent an alle DALI Leuchten gesendet.

Objekt:DALI:Leuchte über Kurzadresse ein/aus

Symbol:

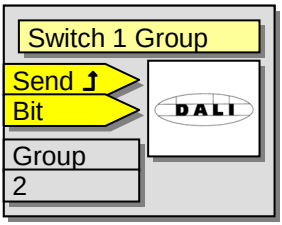


Funktion: Bei steigender Flanke am Eingang Send wird die Leuchte in Abhängigkeit des Bits Bit ein/ausgeschaltet. Betroffen ist nur

die Leuchte mit der Kurzadresse ShortAdr (zB: 1). Als Kurzadresse ist 0..63 zulässig.

Objekt:DALI:Leuchtegruppe ein/aus

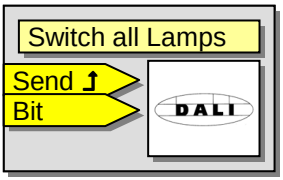
Symbol:



Funktion: Bei steigender Flanke am Eingang Send werden alle Leuchten in Abhängigkeit des Bits Bit ein/ausgeschaltet. Betroffen sind alle Leuchten, welche der Gruppe Group (zB: 2) angehören. Als Gruppe ist 0..15 zulässig.

Objekt:DALI:Alle Leuchten ein/aus

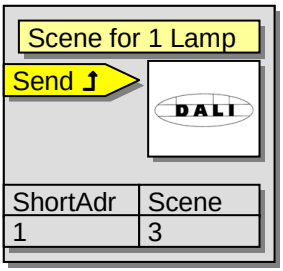
Symbol:



Funktion: Bei steigender Flanke am Eingang Send werden alle Leuchten in Abhängigkeit des Bits Bit ein/ausgeschaltet. Betroffen sidn alle Leuchten, welche am DALI Bus angehängt sind.

Objekt:DALI:Szene für Kurzadresse abrufen

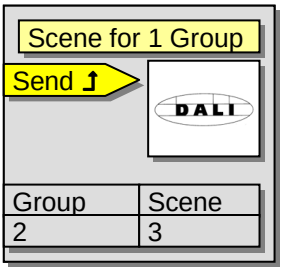
Symbol:



Funktion: Bei steigender Flanke am Eingang Send wird in der Leuchte mit der Kurzadresse ShortAdr (zB: 1) die Szene Scene (zB:3) aufgerufen. Die Leuchte nimmt die für die Szene 3 hinterlegte Helligkeit an. Als Szenen sind 0..15 und als Kurzadresse 0..63 zulässig.

Objekt:DALI:Szene für Gruppe abrufen

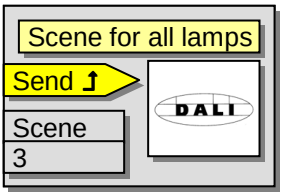
Symbol:



Funktion: Bei steigender Flanke am Eingang Send wird für die Leuchten der Gruppe Group (zB: 2) die Szene Scene (zB:3) aufgerufen. Die Leuchten nehmen die für die Szene 3 hinterlegten Helligkeitswerte an. Als Szenen sind 0..15 und als Gruppe 0..15 zulässig.

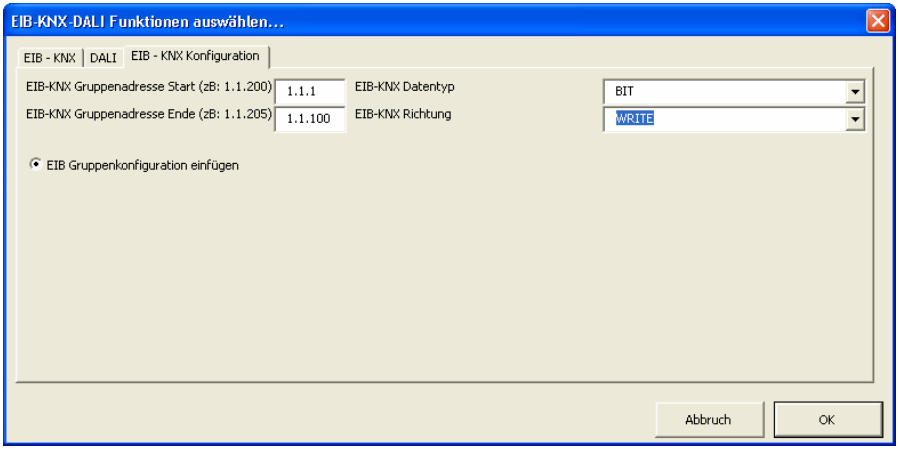
Objekt:DALI:Szene für alle Leuchten abrufen

Symbol:



Funktion: Bei steigender Flanke am Eingang Send wird für alle Leuchten die Szene Scene (zB:3) aufgerufen. Die Leuchten nehmen die für die Szene 3 hinterlegten Helligkeitswerte an. Als Szene ist 0..15 zulässig.

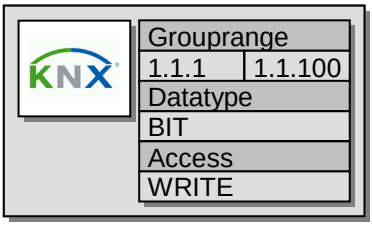
EIB/KNX Konfiguration



Unter dieser Funktionsgruppe sind alle EIB Konfigurationsbefehle zusammengefasst.

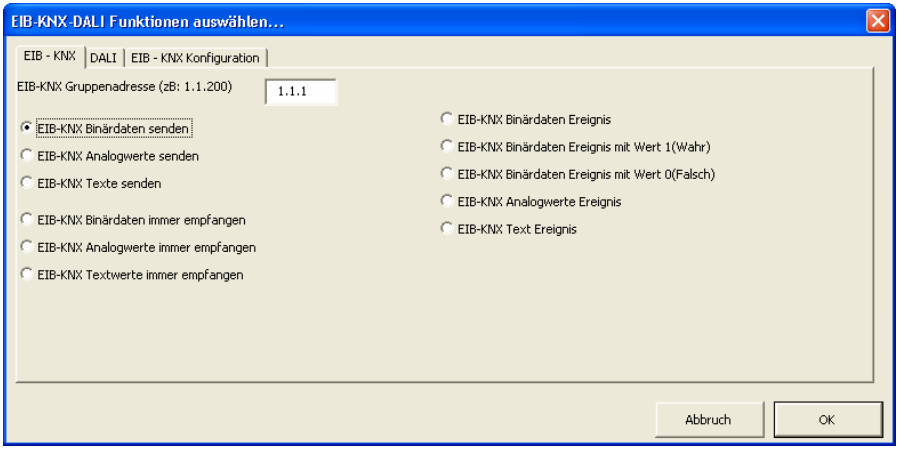
Objekt:EIB/KNX:Gruppenadressen konfiguieren

Symbol:



Funktion: Dieser Befehl legt für den EIB Bus fest, dass die Gruppenadressen im Bereich Grouprange (zB: 1.1.1 bis 1.1.100) den Datentyp Datatype (zB: BIT) verwenden und welche Zugriffsrechte der Controller hat (Access zB: WRITE). Jede EIB/KNX Gruppenadresse muss vorher mit dieser Funktion definiert werden, bevor diese eingelesen oder gesendet werden kann. Es stehen alle EIS Datentypen zur Verfügung. Als Access gibt es READ, WRITE oder READWRITE.

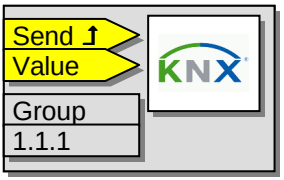
EIB/KNX Verwendung



Unter dieser Funktionsgruppe sind alle EIB Befehle zusammengefasst. Jeder Befehl wirkt auf eine Gruppenadresse, die zuerst eingegeben werden muss.

Objekt:EIB/KNX:Binärdaten senden

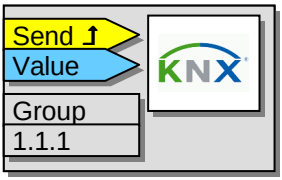
Symbol:



Funktion: Dieser Befehl versendet bei steigender Flanke am Eingang Send in Abhängigkeit des Wertes Value die Gruppenadresse Group. Ist Value gleich 0 so wird die zB: 1.1.1=0 am EIB Bus gesendet. Ist Value ein, so wird zB: 1.1.1=1 gesendet.

Objekt:EIB/KNX:Analogwerte senden

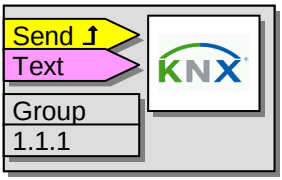
Symbol:



Funktion: Dieser Befehl versendet bei steigender Flanke am Eingang Send den Analogwert Value mit der Gruppenadresse Group über den EIB Bus. Beispiel: 1.1.1=Value

Objekt:EIB/KNX:Textwerte senden

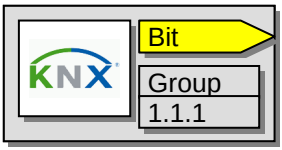
Symbol:



Funktion: Dieser Befehl versendet bei steigender Flanke am Eingang Send den Textwert Value mit der Gruppenadresse Group über den EIB Bus. Beispiel: 1.1.1=Text

Objekt:EIB/KNX:Binärdaten immer empfangen

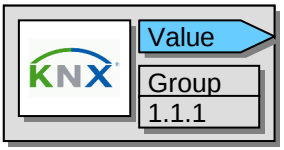
Symbol:



Funktion: Dieser Befehl legt einen internen Bitspeicher an. Wird nun die EIB Gruppe Group (zB: 1.1.1) empfangen, so wird der aktuelle Wert des EIB Telegramms in den internen Speicher übertragen. Das Objekt gibt bei jedem Zyklus den Wert des internen Speichers am Bitausgang Bit aus. Nach dem Neustart hat der interne Speicher immer den Wert 0!

Objekt:EIB/KNX:Analogdaten immer empfangen

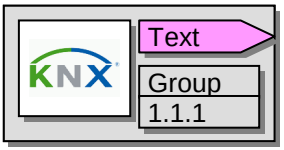
Symbol:



Funktion: Dieser Befehl legt einen internen Analogspeicher an. Wird nun die EIB Gruppe Group (zB: 1.1.1) empfangen, so wird der aktuelle Wert des EIB Telegramms in den internen Speicher übertragen. Das Objekt gibt bei jedem Zyklus den Wert des internen Speichers am Analogausgang Value aus. Nach dem Neustart hat der interne Speicher immer den Wert 0!

Objekt:EIB/KNX:Textdaten immer empfangen

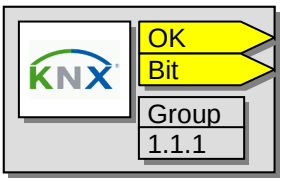
Symbol:



Funktion: Dieser Befehl legt einen internen Textspeicher an. Wird nun die EIB Gruppe Group (zB: 1.1.1) empfangen, so wird der aktuelle Wert des EIB Telegramms in den internen Speicher übertragen. Das Objekt gibt bei jedem Zyklus den Wert des internen Speichers am Textausgang Text aus. Nach dem Neustart hat der interne Speicher immer den Wert 0!

Objekt:EIB/KNX:Binärdaten Ereignis

Symbol:

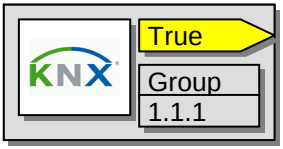


Funktion: Wann immer am EIB Bus ein Telegramm beim Controller ankommt, wird dieses Telegramm in einen internen Ringpuffer übertragen. Pro RESI GRAPH Zyklus wird immer nur ein empfangenes EIB Telegramm aus diesem Ringpuffer ausgelesen und verarbeitet. Man kann nun auf das Ankommen eines EIB Telegramms direkt reagieren. Der Ausgang OK wird

exakt für einen Zyklus eins, wenn das EIB Telegramm gerade angekommen ist. Gleichzeitig steht am Ausgang Bit auch nur für einen Zyklus der aktuelle Datenwert der EIB Gruppeadresse Group (zB: 1.1.1) an.

Objekt:EIB/KNX:Binärdaten Ereignis mit Wert 1 (true)

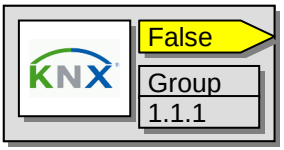
Symbol:



Funktion: Wann immer am EIB Bus ein Telegramm beim Controller ankommt, wird dieses Telegramm in einen internen Ringpuffer übertragen. Pro RESI GRAPH Zyklus wird immer nur ein empfangenes EIB Telegramm aus diesem Ringpuffer ausgelesen und verarbeitet. Man kann nun auf das Ankommen eines EIB Telegramms direkt reagieren. Der Ausgang True wird exakt für einen Zyklus eins, wenn das EIB Telegramm gerade angekommen ist und dessen Gruppenadresse mit der in Group konfigurierten Adresse zusammenstimmt und die Daten des EIB Telegramms 1 sind.

Objekt:EIB/KNX:Binärdaten Ereignis mit Wert 0 (false)

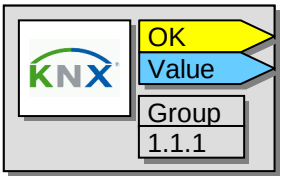
Symbol:



Funktion: Wann immer am EIB Bus ein Telegramm beim Controller ankommt, wird dieses Telegramm in einen internen Ringpuffer übertragen. Pro RESI GRAPH Zyklus wird immer nur ein empfangenes EIB Telegramm aus diesem Ringpuffer ausgelesen und verarbeitet. Man kann nun auf das Ankommen eines EIB Telegramms direkt reagieren. Der Ausgang True wird exakt für einen Zyklus eins, wenn das EIB Telegramm gerade angekommen ist und dessen Gruppenadresse mit der in Group konfigurierten Adresse zusammenstimmt und die Daten des EIB Telegramms 0 sind.

Objekt:EIB/KNX:Analogdaten Ereignis

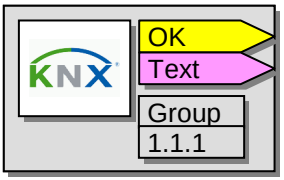
Symbol:



Funktion: Wann immer am EIB Bus ein Telegramm beim Controller ankommt, wird dieses Telegramm in einen internen Ringpuffer übertragen. Pro RESI GRAPH Zyklus wird immer nur ein empfangenes EIB Telegramm aus diesem Ringpuffer ausgelesen und verarbeitet. Man kann nun auf das Ankommen eines EIB Telegramms direkt reagieren. Der Ausgang OK wird exakt für einen Zyklus eins, wenn das EIB Telegramm gerade angekommen ist. Gleichzeitig steht am Ausgang Value auch nur für einen Zyklus der aktuelle Datenwert der EIB Gruppeadresse Group (zB: 1.1.1) an.

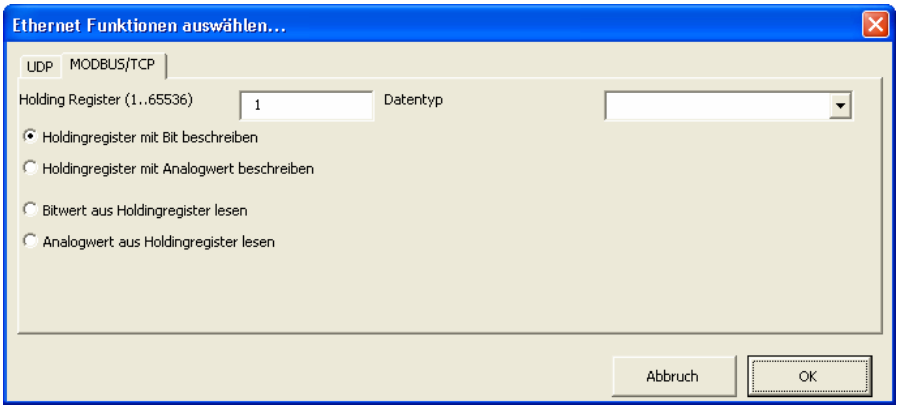
Objekt:EIB/KNX:Textdaten Ereignis

Symbol:



Funktion: Wann immer am EIB Bus ein Telegramm beim Controller ankommt, wird dieses Telegramm in einen internen Ringpuffer übertragen. Pro RESI GRAPH Zyklus wird immer nur ein empfangenes EIB Telegramm aus diesem Ringpuffer ausgelesen und verarbeitet. Man kann nun auf das Ankommen eines EIB Telegramms direkt reagieren. Der Ausgang OK wird exakt für einen Zyklus eins, wenn das EIB Telegramm gerade angekommen ist. Gleichzeitig steht am Ausgang Text auch nur für einen Zyklus der aktuelle Datenwert der EIB Gruppeadresse Group (zB: 1.1.1) an.

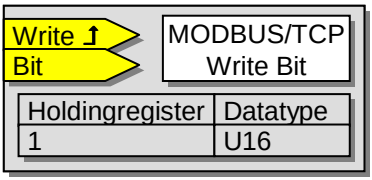
MODBUS/TCP Server



Unter dieser Funktionsgruppe sind alle Befehle zusammengefasst, welche den internen MODBUS/TCP Server betreffen.

Objekt:MODBUS/TCP:Holdingregister mit Bit beschreiben

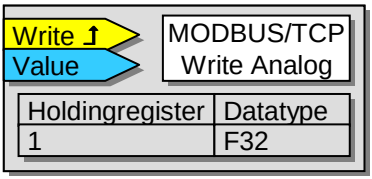
Symbol:



Funktion: Liegt eine steigende Flanke am Eingang Write an, so wird der Bitwert Bit in das Holdingregister mit dem Datentyp Datatype geschrieben. Als Datentypen stehen U16 für Unsigned 16 Bit, U32 für Unsigned 32 Bit und F32 für Float 4 Byte zur Verfügung.

Objekt:MODBUS/TCP:Holdingregister mit Analogwert beschreiben

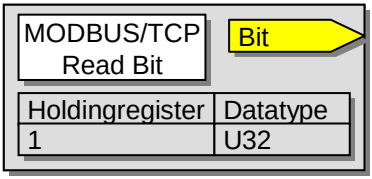
Symbol:



Funktion: Liegt eine steigende Flanke am Eingang Write an, so wird der Analogwert Value in das Holdingregister mit dem Datentyp Datatype geschrieben. Als Datentypen stehen U16 für Unsigned 16 Bit, U32 für Unsigned 32 Bit und F32 für Float 4 Byte zur Verfügung.

Objekt:MODBUS/TCP:Bitwert aus Holdingregister lesen

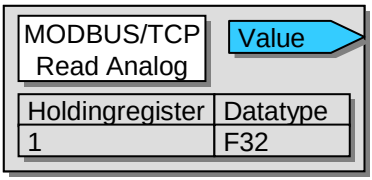
Symbol:



Funktion: Dieser Baustein stellt den Inhalt des Holdingregisters als Bitwert zur Verfügung. Dazu wird das Holdingregister Holdingregister unter der Berücksichtigung des Datentyps Datentyp gelesen. Ist der Inhalt ungleich 0, so wird der Bitwert 1 zurückgegeben, sonst 0.

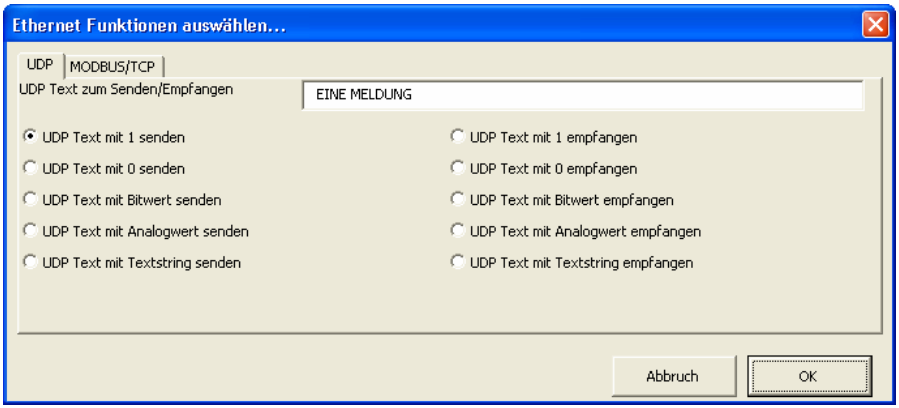
Objekt:MODBUS/TCP:Analogwert aus Holdingregister lesen

Symbol:



Funktion: Dieser Baustein stellt den Inhalt des Holdingregisters als Analogwert zur Verfügung. Dazu wird das Holdingregister Holdingregister unter der Berücksichtigung des Datentyps Datentyp gelesen. ACHTUNG: Der Analogwert im Holdingregister kann den Zahlenbereich des RESI GRAPH Interpreters überschreiten.

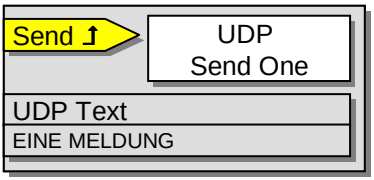
UDP Socket



In der SPS gibt es einen UDP Sockettreiber, der immer das UDP Port 1030 verwendet. Über diesen UDP Sockettreiber können UDP Pakete versendet und empfangen werden. UDP ist ein Protokoll, welches Broadcast unterstützt. Das bedeuted: Wenn ein Teilnehmer ein UDP Paket versendet , empfangen alle anderen Teilnehmer das UDP Paket. Es kann aber auch sein, das ein UDP Paket verloren geht!

Objekt:UDP :Nachricht mit Datenwert 1 senden

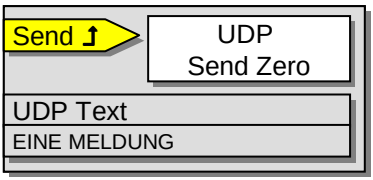
Symbol:



Funktion: Bei steigender Flanke wird über das UDP Port der Text „EINE MELDUNG“ mit einer abschließenden ASCII 0 gesendet. Intern wird die Meldung in folgendes Format umgewandelt: „RESI:EINE MELDUNG=1“ 0 (Abschließende Null)

Objekt:UDP :Nachricht mit Datenwert 0 senden

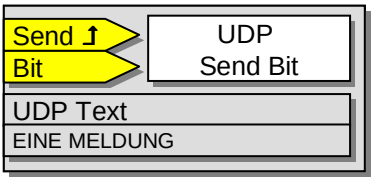
Symbol:



Funktion: Bei steigender Flanke wird über das UDP Port der Text „EINE MELDUNG“ mit einer abschließenden ASCII 0 gesendet. Intern wird die Meldung in folgendes Format umgewandelt: „RESI:EINE MELDUNG=0“ 0 (Abschließende Null)

Objekt:UDP :Nachricht mit Bitwert senden

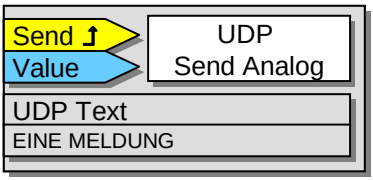
Symbol:



Funktion: Bei steigender Flanke wird über das UDP Port der Text „EINE MELDUNG“ mit einer abschließenden ASCII 0 gesendet. Der Datenwert am Eingang Bit wird mit übertragen. Intern wird die Meldung in folgendes Format umgewandelt: „RESI:EINE MELDUNG=<Bit>“ 0 (Abschließende Null)

Objekt:UDP :Nachricht mit Analogwert senden

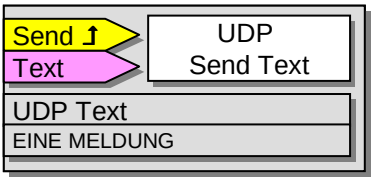
Symbol:



Funktion: Bei steigender Flanke wird über das UDP Port der Text „EINE MELDUNG“ mit einer abschließenden ASCII 0 gesendet. Der Datenwert am Eingang Value wird mit übertragen. Intern wird die Meldung in folgendes Format umgewandelt: „RESI:EINE MELDUNG=<Value>“ 0 (Abschließende Null)

Objekt:UDP :Nachricht mit Textwert senden

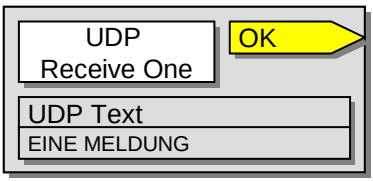
Symbol:



Funktion: Bei steigender Flanke wird über das UDP Port der Text „EINE MELDUNG“ mit einer abschließenden ASCII 0 gesendet. Der Textwert am Eingang Text wird mit übertragen. Intern wird die Meldung in folgendes Format umgewandelt: „RESI:EINE MELDUNG=“<Text>““ 0 (Abschließende Null)

Objekt:UDP :Nachricht mit Bitwert 1 enpfangen

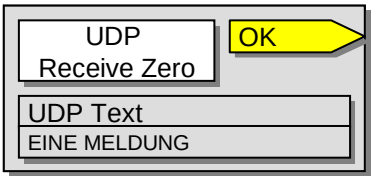
Symbol:



Funktion: Wird über das UDP Port der Text „EINE MELDUNG“ empfangen, so wird der gesendete Wert interpretiert. Ist dieser Wert 1 und der Nachrichtentext stimmt mit dem konfigurierten UDP Text zusammen, so wird für einen Zyklus der Ausgang OK eins. Intern muss die Meldung in folgendem Format angekommen sein: „RESI:EINE MELDUNG=1“ 0 (Abschließende Null)

Objekt:UDP :Nachricht mit Bitwert 0 enpfangen

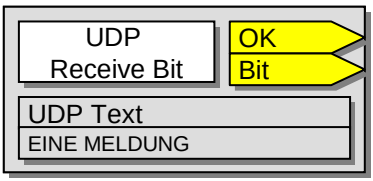
Symbol:



Funktion: Wird über das UDP Port der Text „EINE MELDUNG“ empfangen, so wird der gesendete Wert interpretiert. Ist dieser Wert 0 und der Nachrichtentext stimmt mit dem konfigurierten UDP Text zusammen, so wird für einen Zyklus der Ausgang OK eins. Intern muss die Meldung in folgendem Format angekommen sein: „RESI:EINE MELDUNG=0“ 0 (Abschließende Null)

Objekt:UDP :Nachricht mit Bitwert enpfangen

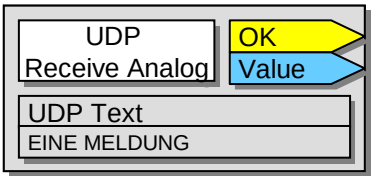
Symbol:



Funktion: Wird über das UDP Port der Text „EINE MELDUNG“ empfangen, so wird der gesendete Wert in einen Bitwert umgewandelt. Stimmt der Nachrichtentext mit dem konfigurierten UDP Text zusammen, so wird für einen Zyklus der Ausgang OK eins und am Ausgang Bit wird der gesendete Wert ausgegeben (Ebenfalls für nur einen Zyklus). Intern muss die Meldung in folgendem Format angekommen sein: „RESI:EINE MELDUNG=<Bit>“ 0 (Abschließende Null)

Objekt:UDP :Nachricht mit Analogwert enpfangen

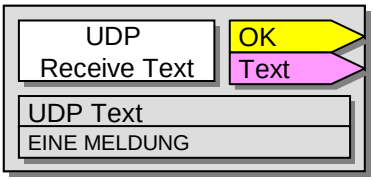
Symbol:



Funktion: Wird über das UDP Port der Text „EINE MELDUNG“ empfangen, so wird der gesendete Wert in einen Analogwert umgewandelt. Stimmt der Nachrichtentext mit dem konfigurierten UDP Text zusammen, so wird für einen Zyklus der Ausgang OK eins und am Ausgang Value wird der gesendete Wert ausgegeben (Ebenfalls für nur einen Zyklus). Intern muss die Meldung in folgendem Format angekommen sein: „RESI:EINE MELDUNG=<Value>“ 0 (Abschließende Null)

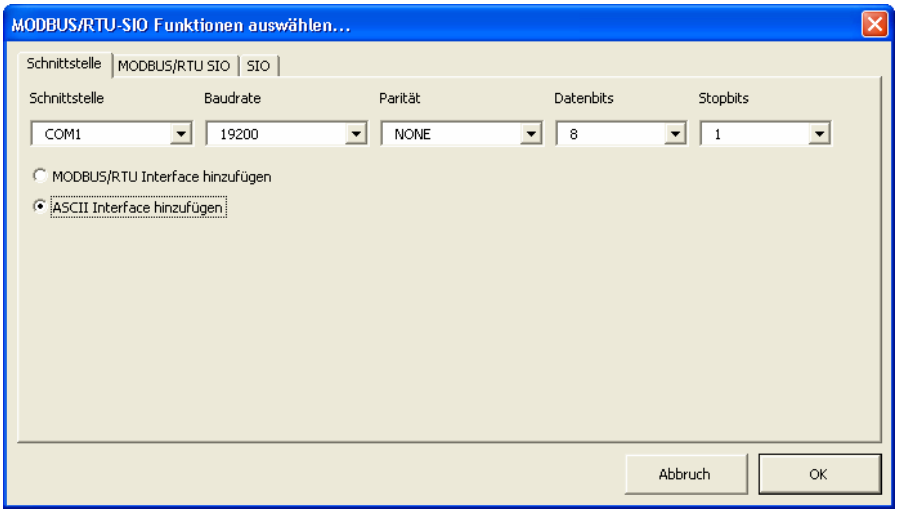
Objekt:UDP :Nachricht mit Textwert enpfangen

Symbol:



Funktion: Wird über das UDP Port der Text „EINE MELDUNG“ empfangen, so wird der gesendete Wert in einen Analogwert umgewandelt. Stimmt der Nachrichtentext mit dem konfigurierten UDP Text zusammen, so wird für einen Zyklus der Ausgang OK eins und am Ausgang Text wird der gesendete Text ausgegeben (Ebenfalls für nur einen Zyklus). Intern muss die Meldung in folgendem Format angekommen sein: „RESI:EINE MELDUNG=“<Text>““ 0 (Abschließende Null)

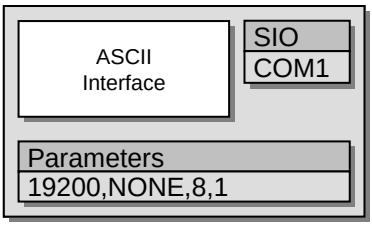
ASCII Schnittstellen



In der SPS kann man freie serielle Schnittstellen auch als ASCII Schnittstellen konfigurieren. Damit kann man dann mit Multimediageräten eine Kommunikation aufbauen oder serielle Drucker mit Daten beschicken oder Barcodereader einlesen. Für komplexe Protokolle ist diese Funktion aber nicht gedacht!

Objekt:ASCII :Schnittstelle definieren

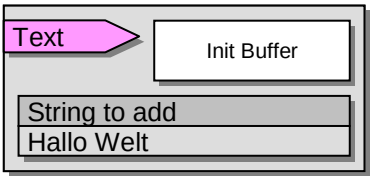
Symbol:



Funktion: Mit diesem Objekt wird eine freie serielle Schnittstelle als ASCII Schnittstelle festgelegt. Unter Parameters stehen die Schnittstellendaten für die Übertragungsgeschwindigkeit und das Datenformat.

Objekt:ASCII :Sendebuffer initialisieren

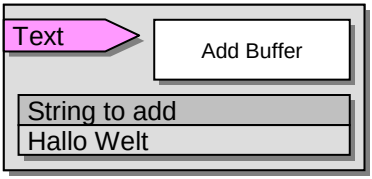
Symbol:



Funktion: Dieser Befehl löscht zuerst den internen Sendebuffer. Danach wird der Inhalt des Wertes String to add in den Sendebuffer übertragen. Wurde der Eingang Text verdrahtet, so wird der übergebene Text auch dem Sendebuffer hinzugefügt.

Objekt:ASCII :Text dem Sendebuffer hinzufügen

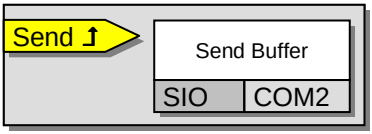
Symbol:



Funktion: Dieser Befehl fügt den Inhalt des Wertes String to add an den Sendebuffer an.

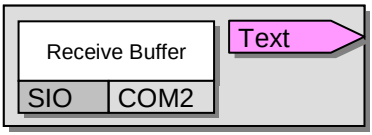
Objekt:ASCII :Sendebuffer senden

Symbol:



Funktion: Dieser Befehl sendet den Inhalt des Sendebuffers an die serielle Schnittstelle SIO. Wichtig ist, es gibt nur einen internen Sendebuffer für alle seriellen Schnittstellen.

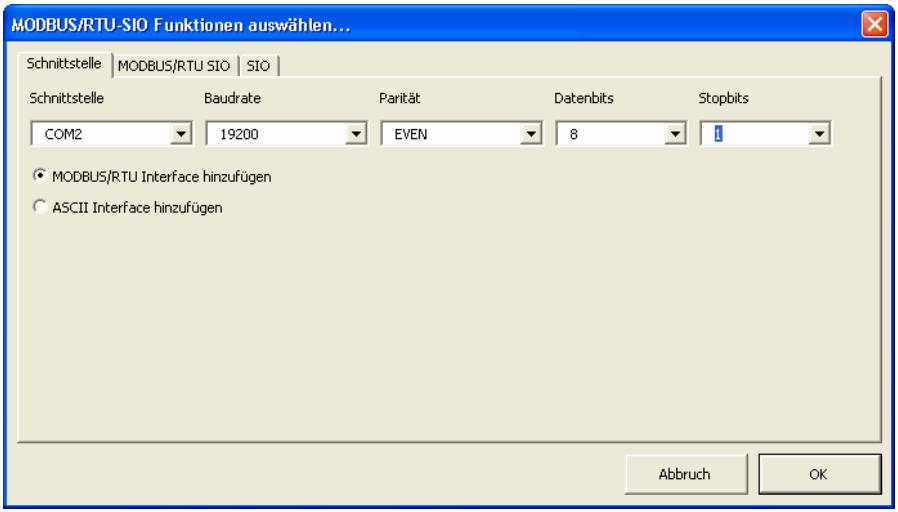
Objekt:ASCII :Empfangsbuffer auslesen



Symbol:

Funktion: Dieser Befehl prüft, ob der interne Empfangbuffer einen Wert enthält. Ein Wert ist eine Zeichenkette, welche entweder mit 0x00 oder mit 0x0D abgeschlossen wurde. Dann wird exakt für einen Zyklus der Textwert zurückgegeben. ACHTUNG: RESI GRAPH kann nur Texte mit einer Maximallänge von 20 Zeichen verarbeiten. Ist der Buffer leer oder ist noch kein Endezeichen angekommen, so wird ein Leerstring zurückgegeben.

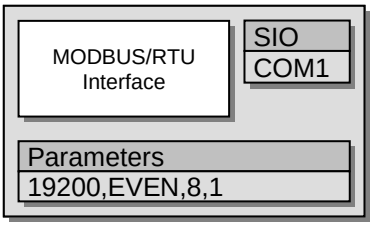
MODBUS/RTU Master



In der SPS kann man freie serielle Schnittstellen auch als MODBUS/RTU Master konfigurieren. Dann werden gewisse MODBUS/RTU Teilnehmer automatisch abgefragt. Diese Funktion ist zur Kommunikation mit Energiezählern oder anderen Geräten gedacht, welche einfache MODBUS/RTU Slaves sind.

Objekt:MODBUS/RTU: Schnittstelle definieren

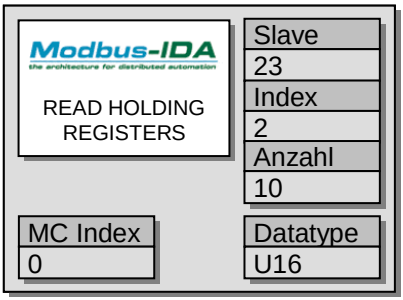
Symbol:



Funktion: Mit diesem Objekt wird eine freie serielle Schnittstelle als MODBUS/RTU Master definiert. Unter Parameters stehen die Schnittstellendaten für die Übertragungsgeschwindigkeit und das Datenformat.

Objekt:MODBUS/RTU: Holdingregister vom Slave lesen

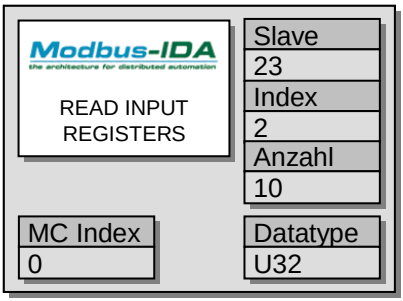
Symbol:



Funktion: Mit diesem Objekt wird in die Abfrageliste für den MODBUS/RTU Master ein Eintrag hinzugefügt, welcher einen MODBUS/RTU Slave zyklisch abfragt und die Daten dann gemäß des spezifizierten Datentyps umwandelt und in die interne MemoryCard ab Adresse MC Index ablegt. Slave steht für die MODBUS/RTU Adresse des abzufragenden Teilnehmers. Index steht für den Startindex des Holdingregisters im Teilnehmer, Anzahl definiert die Anzahl der abzufragenden Holdingregister. Datatype definiert den Datentyp der Information im Register. MC Index definiert den Speicher, ab dem die gelesenen Daten in die Memorycard geschrieben werden sollen.

Objekt:MODBUS/RTU: Inputregister vom Slave lesen

Symbol:

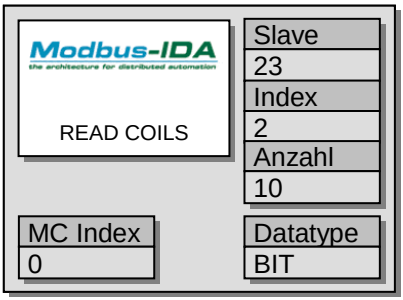


Funktion: Mit diesem Objekt wird in die Abfrageliste für den MODBUS/RTU Master ein Eintrag hinzugefügt, welcher einen MODBUS/RTU Slave zyklisch abfragt und die Daten dann gemäß des spezifizierten Datentyps umwandelt und in die

interne MemoryCard ab Adresse MC Index ablegt. Slave steht für die MODBUS/RTU Adresse des abzufragenden Teilnehmers. Index steht für den Startindex des Inputregisters im Teilnehmer, Anzahl definiert die Anzahl der abzufragenden Inputregister. Datatype definiert den Datentyp der Information im Register. MC Index definiert den Speicher, ab dem die gelesenen Daten in die Memorycard geschrieben werden sollen.

Objekt:MODBUS/RTU: Coils vom Slave lesen

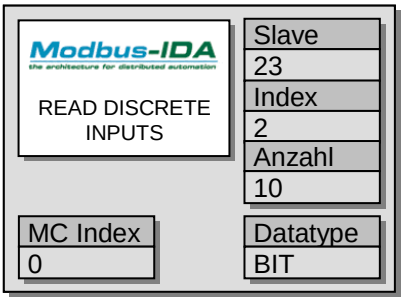
Symbol:



Funktion: Mit diesem Objekt wird in die Abfrageliste für den MODBUS/RTU Master ein Eintrag hinzugefügt, welcher einen MODBUS/RTU Slave zyklisch abfragt und die Daten dann gemäß des spezifizierten Datentyps umwandelt und in die interne MemoryCard ab Adresse MC Index ablegt. Slave steht für die MODBUS/RTU Adresse des abzufragenden Teilnehmers. Index steht für den Startindex der Coils im Teilnehmer, Anzahl definiert die Anzahl der abzufragenden Coils. Datatype definiert den Datentyp der Information in den Coils (immer BIT). MC Index definiert den Speicher, ab dem die gelesenen Daten in die Memorycard geschrieben werden sollen.

Objekt:MODBUS/RTU: Discrete Inputs vom Slave lesen

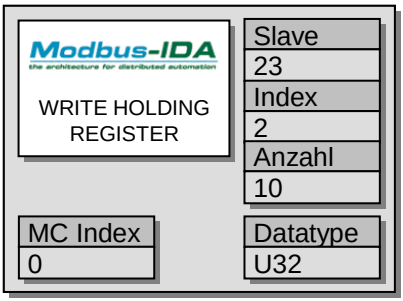
Symbol:



Funktion: Mit diesem Objekt wird in die Abfrageliste für den MODBUS/RTU Master ein Eintrag hinzugefügt, welcher einen MODBUS/RTU Slave zyklisch abfragt und die Daten dann gemäß des spezifizierten Datentyps umwandelt und in die interne MemoryCard ab Adresse MC Index ablegt. Slave steht für die MODBUS/RTU Adresse des abzufragenden Teilnehmers. Index steht für den Startindex der Discrete Inputs im Teilnehmer, Anzahl definiert die Anzahl der abzufragenden Discrete Inputs. Datatype definiert den Datentyp der Information in den Discrete Inputs (immer BIT). MC Index definiert den Speicher, ab dem die gelesenen Daten in die Memorycard geschrieben werden sollen.

Objekt:MODBUS/RTU: Holding Register in den Slave schreiben

Symbol:

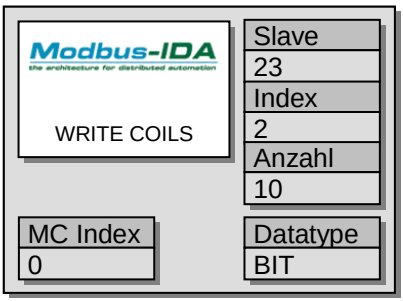


Funktion: Mit diesem Objekt wird in die Abfrageliste für den MODBUS/RTU Master ein Eintrag hinzugefügt, welcher einen MODBUS/RTU Slave zyklisch mit Daten beschreibt. Diese Daten werden zuerst aus der Memorycard ab dem Index

MCIndex ausgelesen und dann gemäß des angegebenen Datentyps umgewandelt und in die Holdingregister des Slaves geschrieben. Slave steht für die MODBUS/RTU Adresse des abzufragenden Teilnehmers. Index steht für den Startindex der Holding Register im Teilnehmer, Anzahl definiert die Anzahl der zu schreibenden Register. Datatype definiert den Datentyp der Information in den Holdingregistern (immer BIT). MC Index definiert den Speicher, ab dem die Daten aus der Memorycard gelesen werden sollen.

Objekt:MODBUS/RTU: Coils in den Slave schreiben

Symbol:

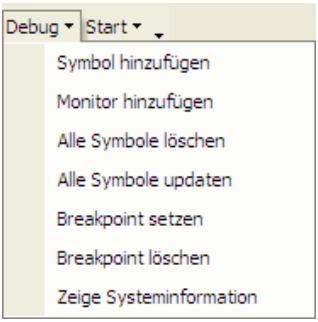


Funktion: Mit diesem Objekt wird in die Abfrageliste für den MODBUS/RTU Master ein Eintrag hinzugefügt, welcher einen MODBUS/RTU Slave zyklisch mit Daten beschreibt. Diese Daten werden zuerst aus der Memorycard ab dem Index MCIndex ausgelesen und dann gemäß des angegebenen Datentyps umgewandelt und in die Coils des Slaves geschrieben. Slave steht für die MODBUS/RTU Adresse des abzufragenden Teilnehmers. Index steht für den Startindex der Coils im Teilnehmer, Anzahl definiert die Anzahl der zu schreibenden Coils. Datatype definiert den Datentyp der Information (immer BIT). MC Index definiert den Speicher, ab dem die zu schreibenden Daten aus der Memorycard gelesen werden sollen.

Debug

Unter diesem Menüpunkt sind alle Aktionen zusammengefasst, die sich zum Debugen/Testen des Programms, befassen.

Wenn Sie den Menüpunkt Debug anwählen erscheint folgende Auswahl:



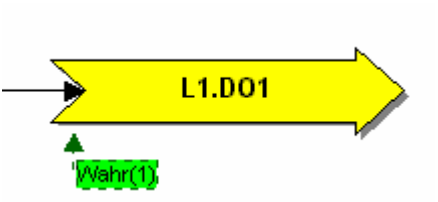
Debug:Symbol hinzufügen

Symbol:

Funktion: Diese Symbol gibt den Zustand einer Variablen oder eines I/Os

Mit Wahr(1) oder Falsch(0) zurück. Jedoch muss man zuerst den Menüpunkt Debug: Alle Symbole updaten durchführen.

Beispiel:



Der Ausgang L1.DO1 ist eins, deshalb Symbol Wahr(1)!

Debug:Alle Symbole löschen

Dieser Menüpunkt löscht alle Symbole auf der aktuellen Seite.

Debug:Alle Symbole updaten

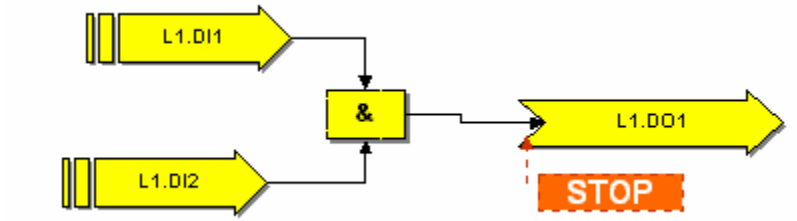
Dieser Menüpunkt ist erforderlich um alle Symbole zu aktualisieren. Ohne Betätigung dieses Punktes können Veränderungen nicht Wahrgenommen werden.

Debug:Breakpoint setzen

Symbol:

Funktion: Programmablauf findet nur bis zu diesem Punkt statt!

Beispiel:



Wenn L1.DO1 eins ist, wird das Programm gestoppt, da der Breakpoint erreicht wurde.

Debug:Breakpoint löschen

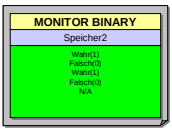
Löscht den Breakpoint aus dem Programm!

Debug:Start mit Breakpoint

Startet dass Programm mit Breakpoint! Programm wird nur so lange abgearbeitet bis der Breakpoint erreicht wird!

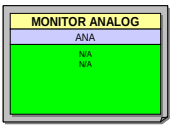
Debug:Monitor hinzufügen

Monitor Binary

Symbol: 

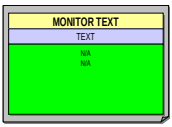
Funktion: Diese Funktion gibt den Zustand einer Variable oder eines I/O mit Wahr(1) oder Falsch(0) zurück. Bevor man diese Funktion nutzen kann, muss man jedoch den Menüpunkt „Alle Symbole updaten" ausführen. Im unterschied zu einem zugewiesenen Symbol kann man mit einem Monitor auch die letzten Zustände sehen.

Monitor Analog

Symbol: 

Funktion: Diese Funktion gibt Ihnen den Zustand einer Analogvariable zurück. Bevor man diese Funktion nutzen kann, muss man jedoch den Menüpunkt „Alle Symbole updaten" ausführen. Im unterschied zu einem zugewiesenen Symbol kann man mit einem Monitor auch die letzten Zustände sehen.

Monitor Text

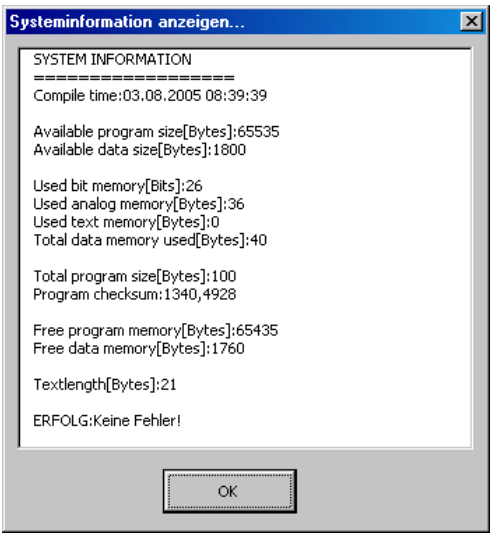
Symbol: 

Funktion: Diese Funktion gibt Ihnen den Zustand einer Textvariable zurück. Bevor man diese Funktion nutzen kann, muss man jedoch den Menüpunkt „Alle Symbole updaten" ausführen. Im unterschied zu einem zugewiesenen Symbol kann man mit einem Monitor auch die letzten Zustände sehen.

!!Wichtig!! Diese drei Monitorfunktionen, können nur ausgeführt werden, wenn eine SPS an Ihren PC angeschlossen ist. Weiters muss das Programm zuerst kompiliert werden, bevor man Symbole hinzufügen bzw. updaten kann.

Debug:Zeige Systeminformationen

Diese Funktion bietet Ihnen die Möglichkeit, wenn Sie ein Projekt kompiliert haben, den Status Ihres Projektes herauszulesen.

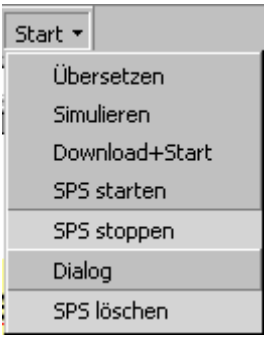


Es wird Ihnen das Datum vom letzten kompilieren, der verwendete Speicher, der freie Speicher, die Checksumme des Projektes und der Status des Kompilierens angezeigt.

Start

Unter diesem Menüpunkt sind alle Aktionen zusammengefasst, die sich mit der Übersetzung, der Simulation und der Ausführung des Programmes innerhalb der SPS befassen.

Wenn Sie den Menüpunkt Start anwählen erscheint folgende Auswahl:



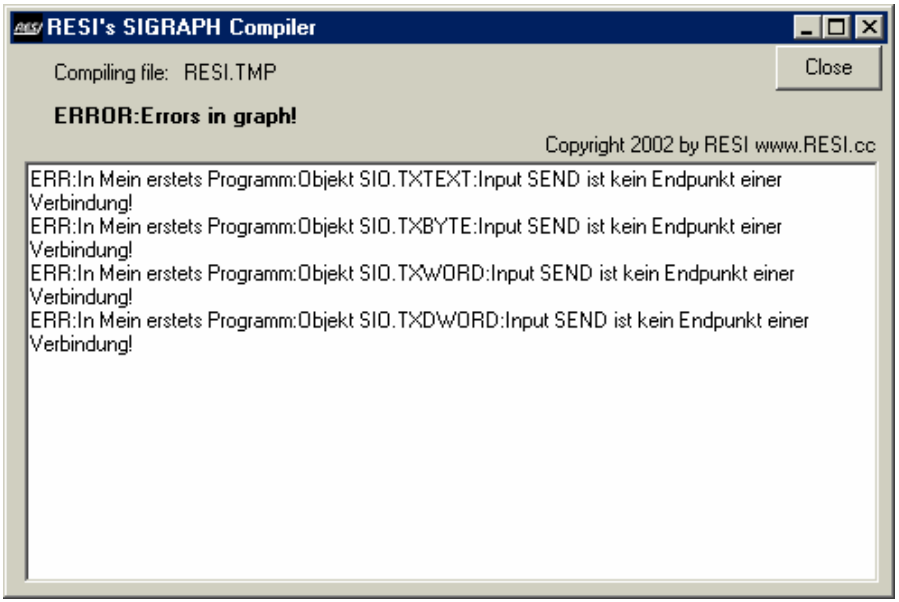
Start:Übersetzen

Dieser Menüpunkt ruft den integrierten Übersetzer auf. Dieser Übersetzer erzeugt aus der aktuellen Grafik ein lauffähiges Programm oder zeigt gewisse Fehler im Plan an.

Statusanzeige des Compilers:

Fehler beim Übersetzen

Wenn Fehler auftreten, werden diese wie folgt dargestellt:


RESI's SIGRAPH Compiler
Compiling file: RESI.TMP
Close
ERROR:Errors in graph!
Copyright 2002 by RESI www.RESI.cc
ERR:In Mein erstes Programm:Objekt SIO.TXTEXT:Input SEND ist kein Endpunkt einer Verbindung!
ERR:In Mein erstes Programm:Objekt SIO.TXBYTE:Input SEND ist kein Endpunkt einer Verbindung!
ERR:In Mein erstes Programm:Objekt SIO.TXWORD:Input SEND ist kein Endpunkt einer Verbindung!
ERR:In Mein erstes Programm:Objekt SIO.TXDWORD:Input SEND ist kein Endpunkt einer Verbindung!

Für jeden Fehler wird der Titel der Seite angegeben, auf der sich der Fehler befindet. Wenn möglich wird auch der exakte Fehlergrund angeführt.

Klicken Sie auf die Schaltfläche Close um den Compiler zu beenden.

Erfolg beim Übersetzen

Hat alles funktioniert, so verschwindet der Compiler wieder vollautomatisch und ein ausführbares Programm steht zur Verfügung.

Start:Simulieren

Sie können ein RESI'S GRAPH Programm auch am Bildschirm simulieren. Wenn das Programm fehlerfrei übersetzt wurde wird automatisch der Simulator gestartet, der folgendes Aussehen hat.

Sart:Download+Start

Wenn Sie diesen Menüpunkt anwählen, wird das Programm automatisch übersetzt. Bei fehlerfreier Übersetzung wird es in die angeschlossene SPS übertragen und dort sofort gestartet.

Sart:SPS starten

Wenn Sie diesen Menüpunkt auswählen, wird das aktuelle SPS Programm neu gestartet.

Sart:SPS stoppen

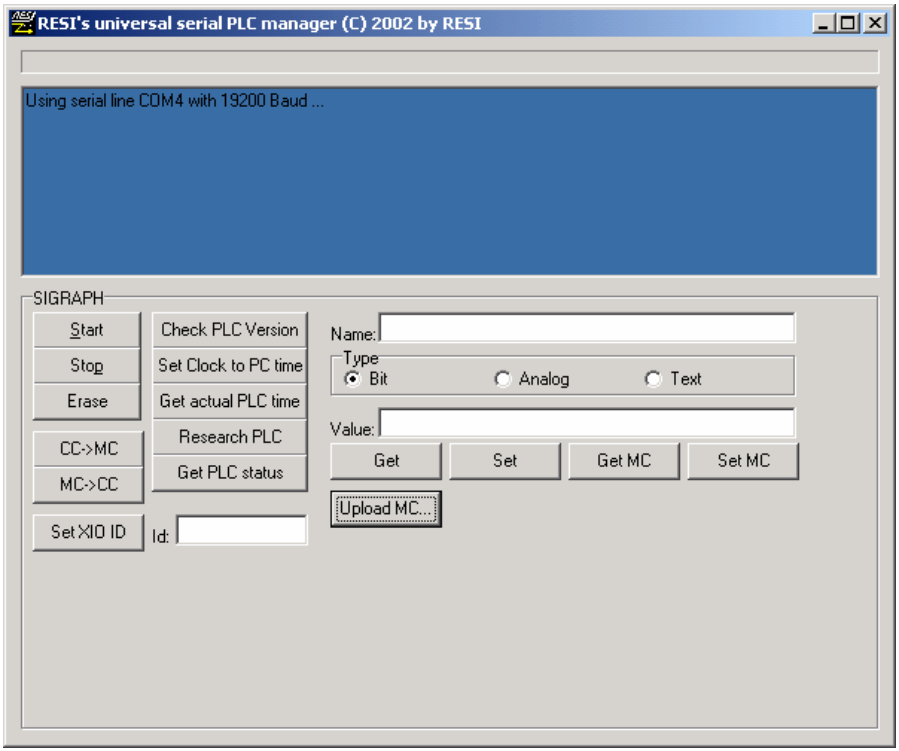
Wenn Sie diesen Menüpunkt auswählen, wird das aktuelle SPS Programm gestoppt.

Sart:SPS löschen

Wenn Sie diesen Menüpunkt auswählen, wird das aktuelle SPS Programm in der SPS gelöscht.

Start:Dialog

Bei Aktivierung dieses Menüpunktes wird der PLC Manager mit einer Oberfläche aktiviert, die besondere Aktionen zulässt:



Schaltfläche:Start

Hiermit kann man das in der SPS vorhandene Programm starten.

Schaltfläche:Stop

Hiermit kann man das in der SPS vorhandene Programm stoppen.

Schaltfläche:Erase

Hiermit kann man das in der SPS vorhandene Programm komplett löschen.

Schaltfläche:CC->MC

Wenn Sie diesen Menüpunkt auswählen, wird das aktuell in der SPS gespeicherte Programm in die eingesteckte MemoryCard übertragen.

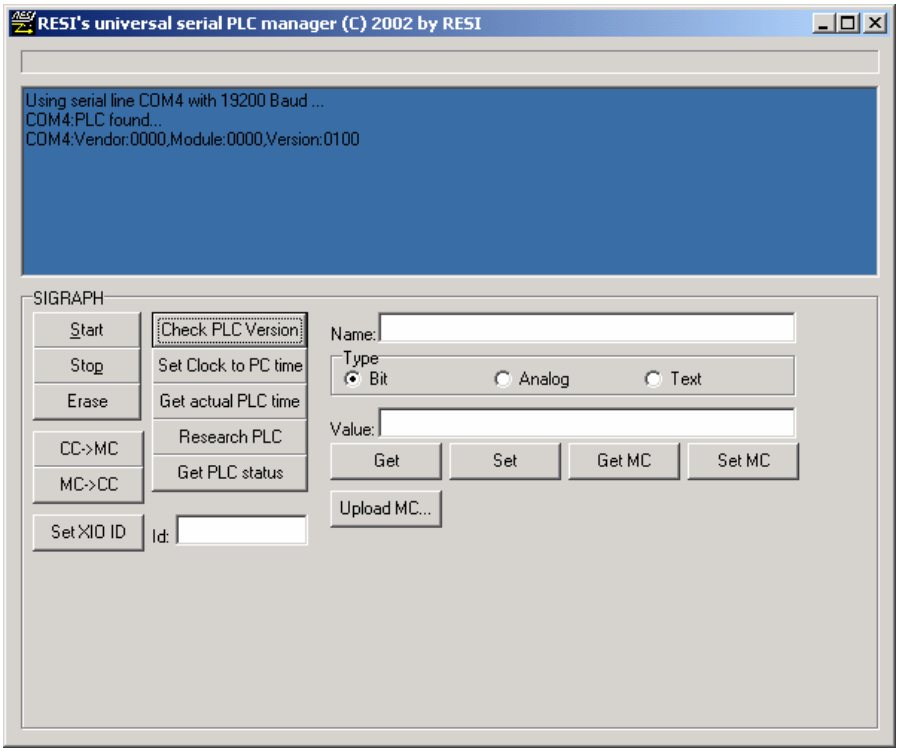
Schaltfläche:MC->CC

Wenn Sie diesen Menüpunkt auswählen, wird das Programm, das in der MemoryCard gespeichert ist, in die SPS übertragen.

Wird eine MemoryCard mit einem gültigen Programm eingesteckt und die SPS neu gestartet, so wird das Programm automatisch in die SPS geladen. Danach kann die MemoryCard entfernt werden.

Schaltfläche:Check PLC Version

Hiermit kann man die Version der Software der angeschlossenen SPS abfragen:

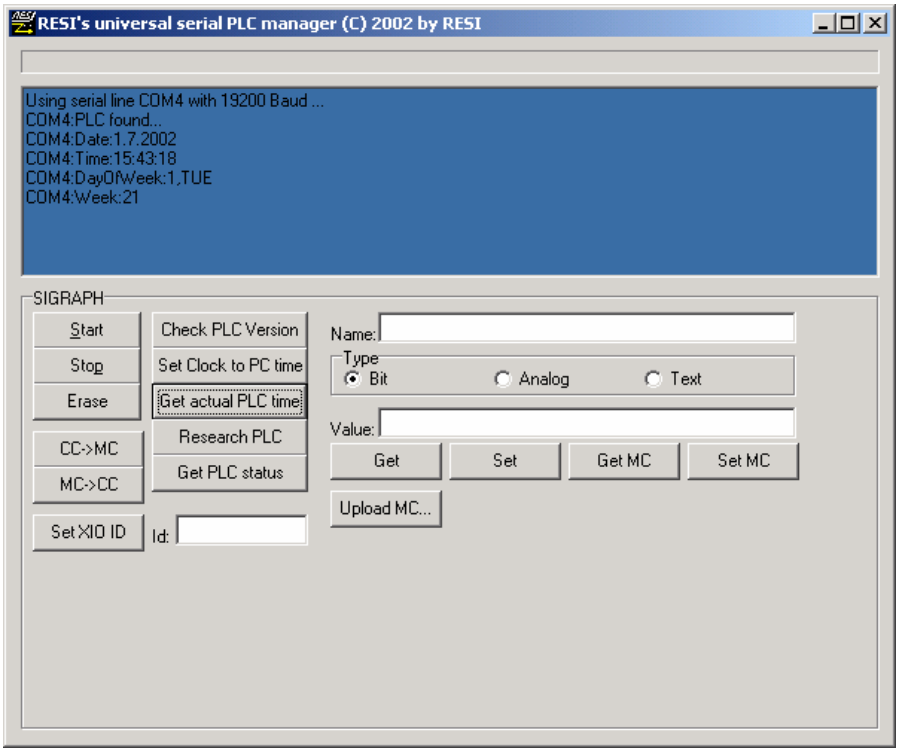


Schaltfläche:Set clock to PC time

Mit dieser Schaltfläche wird die aktuelle Uhrzeit des PCs als neue Zeit für die SPS gesetzt.

Schaltfläche:Get actual PLC time

Mit dieser Schaltfläche kann man die aktuelle Uhrzeit der SPS abfragen und anzeigen.

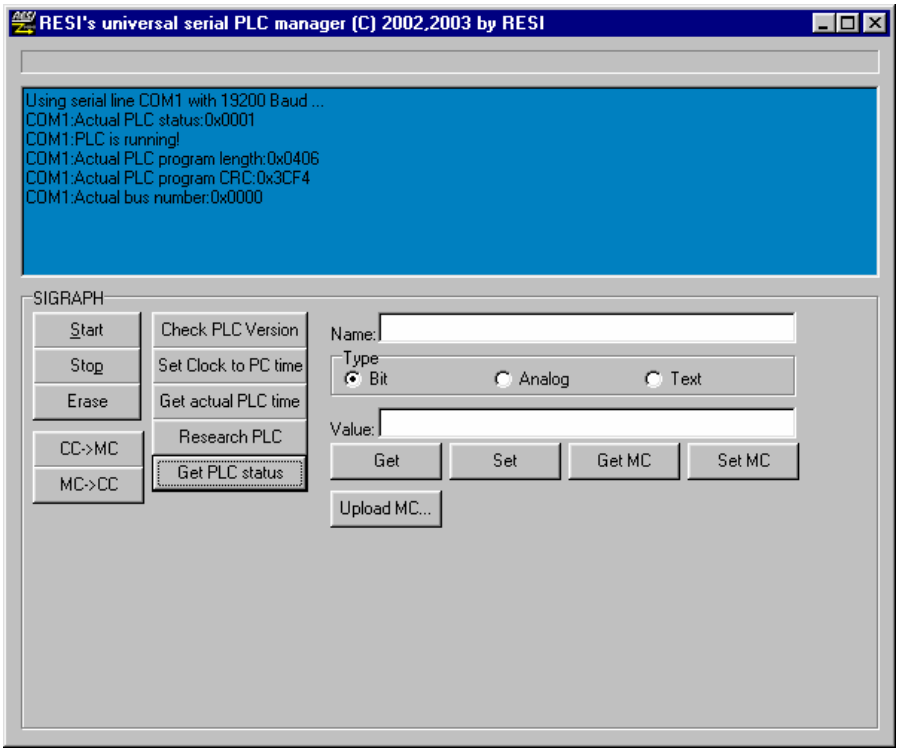


Schaltfläche:Research PLC

Diese Schaltfläche bewirkt das Durchsuchen aller Schnittstellen auf eine angeschlossene SPS. Sieh dazu den Abschnitt SPS nicht gefunden.

Schaltfläche:Get PLC status

Diese Schaltfläche ermöglicht das Abfragen des aktuellen Status der SPS. Angezeigt wird der aktuelle Zustand der SPS (Läuft, Läuft nicht) und die Fehlerzustände. Außerdem sieht man die aktuelle Programmlänge und die Prüfsumme des aktuellen Programme.



Schaltfläche:Upload MC...

Mit dieser Schaltfläche hat man die Möglichkeit die aktuellen Daten der Memory Card in einem Text File auf dem PC zu sichern.

BitSpeicher lesen/schreiben

Der PLC Manager ermöglicht es auch, Bits aus dem laufenden Programm auszulesen bzw. neue Bits zu setzen. Dazu dient der rechte Bereich des PLC Managers.

Unter der Rubrik Name kann man entweder den Namen des Bits eintragen. Der Name entspricht genau dem Namen der Speicher, die in RESI'S GRAPH vergeben wurden. Oder man gibt eine Speichernummer an. Dies geschieht durch

Voranstellen eines # Zeichens vor der Nummer (zum Beispiel #123).

Der Typ muss auf Bit stehen.

Nun können Sie durch Drücken der Schaltfläche Get den aktuellen Zustand des Bits abfragen.

Sie können aber auch in der Rubrik Value einen neuen Wert vorgeben (0 oder 1) und die Schaltfläche Set drücken.

AnalogSpeicher lesen/schreiben

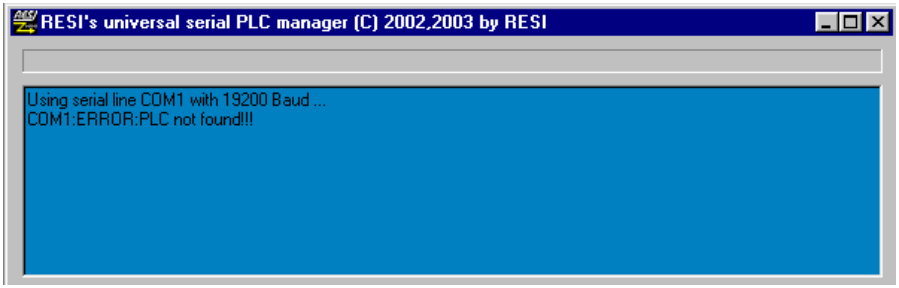
Das Auslesen von Analogwerten erfolgt analog zum Auslesen von Bitwerten. Nur muss der Typ auf Analog stehen. Für das Vorgeben von Analogwerten geben Sie in die Rubrik Value einen gültigen Analogwert ein.

TextSpeicher lesen/schreiben

TextSpeicher können ebenfalls gelesen oder geschrieben werden. Stellen Sie dazu den Typ auf Text und geben Sie unter Value eine Zeichenkette ein.

SPS nicht gefunden

Wird die SPS nicht gefunden oder ist diese nicht angeschlossen, so bleibt der PLC Manager mit folgender Meldung stehen:

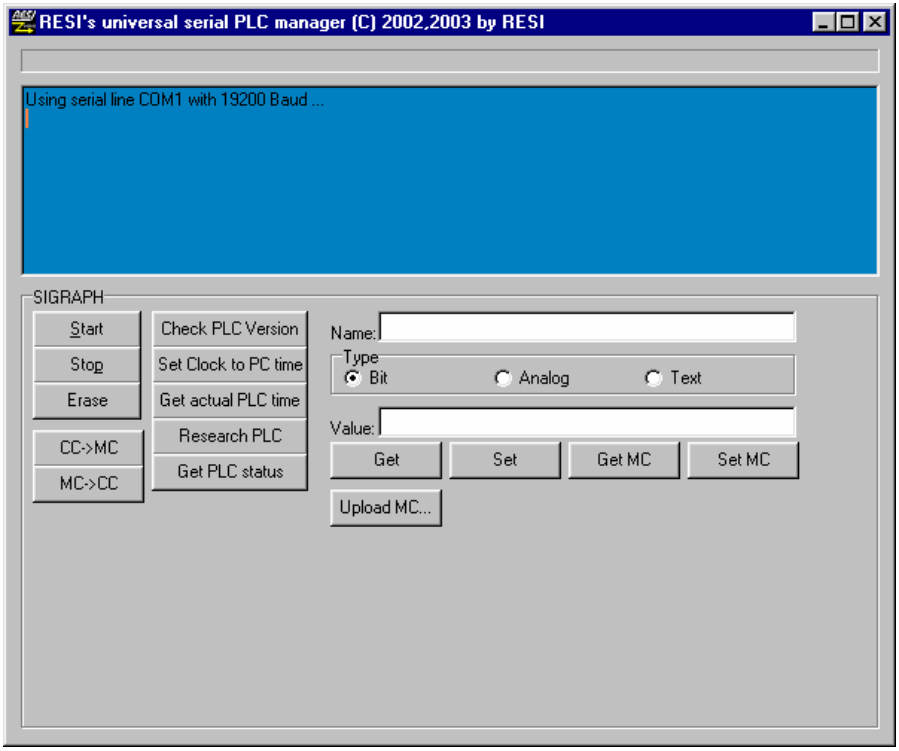


Klicken Sie nun auf das X um den Manager zu beenden. Überprüfen Sie die Verkabelung und die Einstellungen.

Um die serielle Schnittstelle zu wechseln, gehen Sie wie folgt vor:

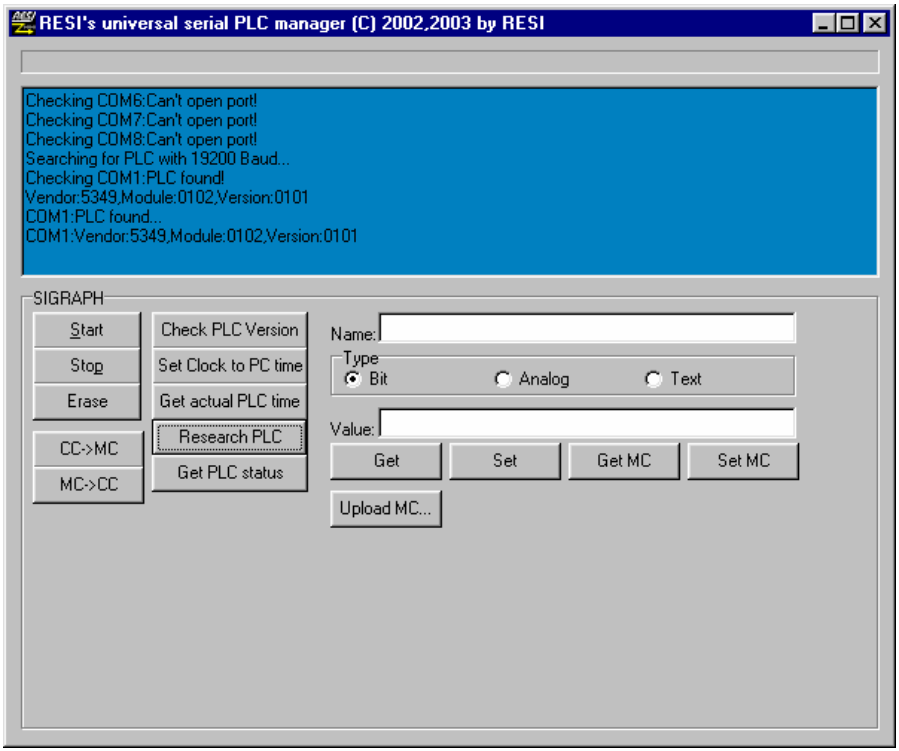
Serielle Schnittstelle auswählen

Wählen Sie den Menüpunkt Start/Dialog an. Es erscheint nach kurzer Zeit der PLC manager mit folgender Oberfläche:



Wenn Sie nun die Schaltfläche Research PLC aktivieren, sucht der PLC manager alle verfügbaren Schnittstellen Ihres Systems ab, um eine SPS zu finden. Ist dies erfolgreich, wird folgende Meldung ausgegeben und die aktuellen Kommunikationsparameter werden hinterlegt.

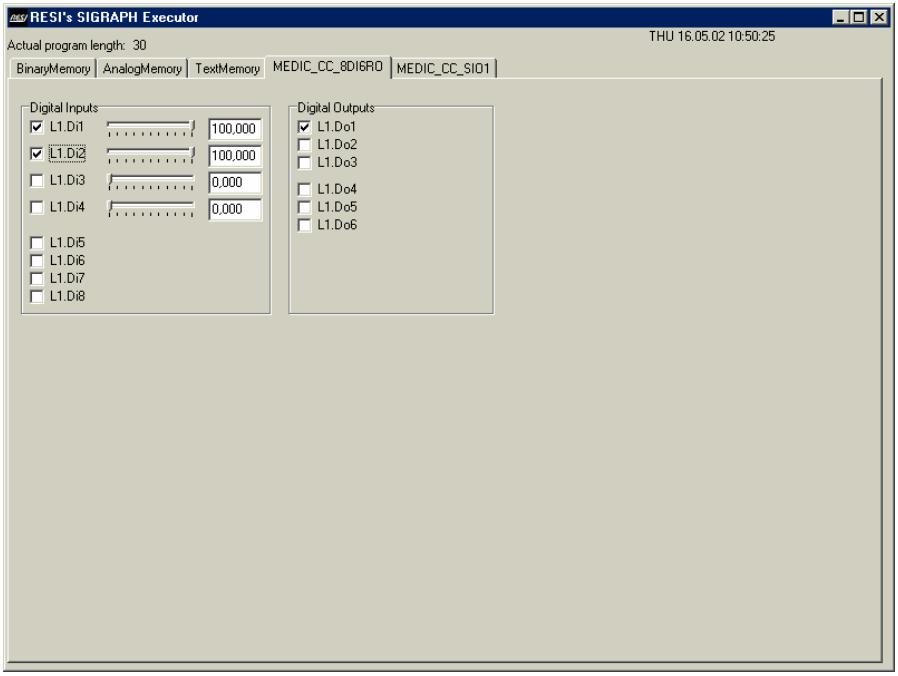
Dies geschieht in einer Datei, die im temporären Verzeichnis von Windows angelegt wird und den Namen RESISETTINGS.TMP trägt. Wenn Sie diese Datei löschen führt der PLC manager bei der nächsten Aktion automatisch eine Suchen nach einer PLC durch.


RESI's universal serial PLC manager (C) 2002,2003 by RESI
Checking COM6:Can't open port!
Checking COM7:Can't open port!
Checking COM8:Can't open port!
Searching for PLC with 19200 Baud...
Checking COM1:PLC found!
Vendor:5349,Module:0102,Version:0101
COM1:PLC found...
COM1:Vendor:5349,Module:0102,Version:0101
SIGRAPH
Start
Stop
Erase
CC->MC
MC->CC
Check PLC Version
Set Clock to PC time
Get actual PLC time
Research PLC
Get PLC status
Name:
Type
Bit
Analog
Text
Value:
Get
Set
Get MC
Set MC
Upload MC...

Simulator

Der integrierte Simulator von RESI'S GRAPH ermöglicht das komplette Testen einer Applikation ohne externe Elemente.

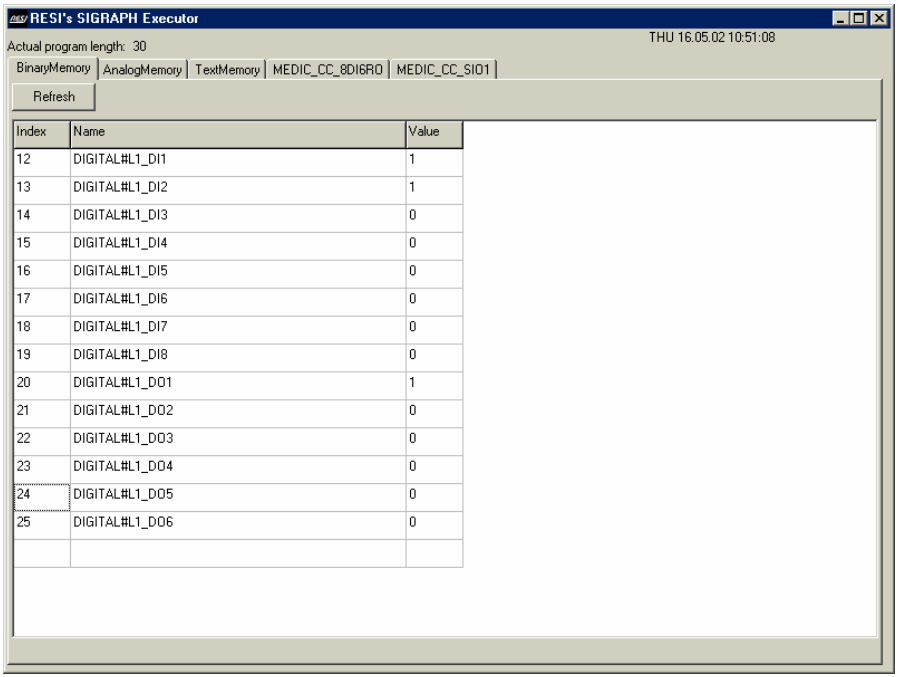
Start des Simulators



Für jedes Element wird eine Simulationsseite eingeblendet.

Simulation:BinaryMemory

Unter dieser Seite werden alle binären Speicher der SPS dargestellt.

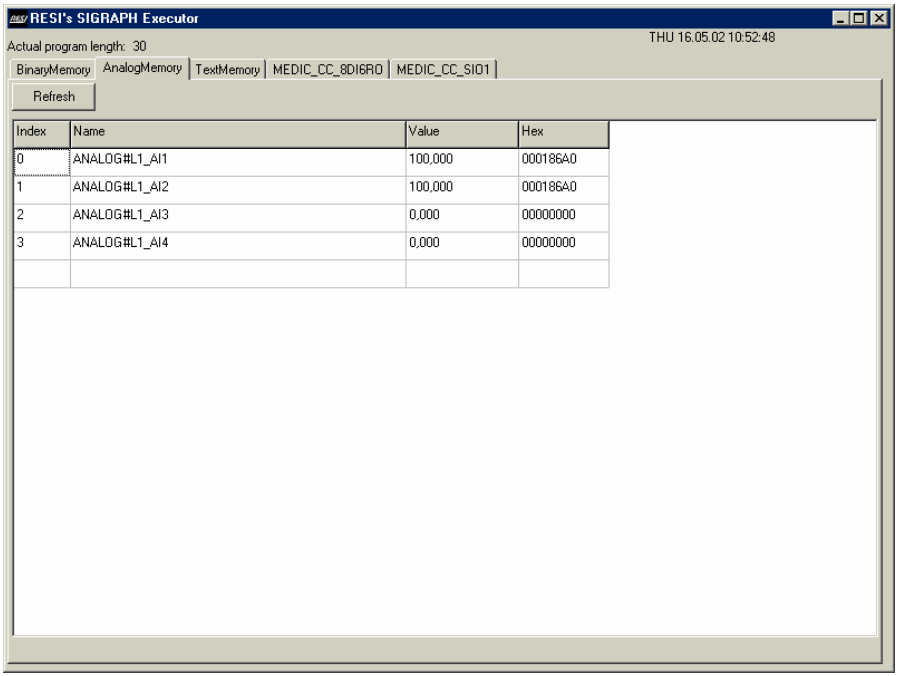


Die Spalte Index bezeichnet den internen Speicherplatz des Merkers. Name beschriebt den Namen und Value beschreibt den aktuellen Wert.

Um den Wert eines Speichers zu ändern, klicken Sie auf dem Feld Index doppelt. Der Wert wird invertiert!

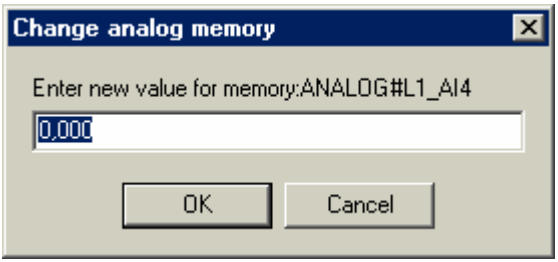
Simulation:AnalogMemory

Auf dieser Seite werden alle Analogwerte dargestellt:



Alle Analogwerte werden sowohl als Analogwert in der Spalte Value als auch als 32-Bit Hexadezimalwert in der Spalte Hex dargestellt.

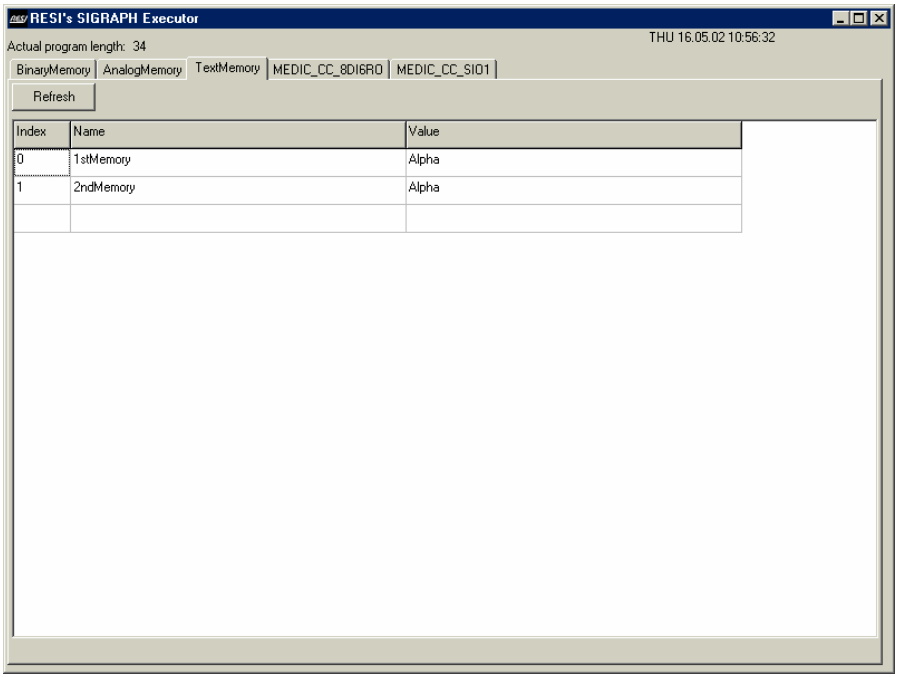
Wollen Sie einen Analogwert ändern, so klicken Sie auf das entsprechende Indexfeld. Es erscheint folgende Eingabemaske:



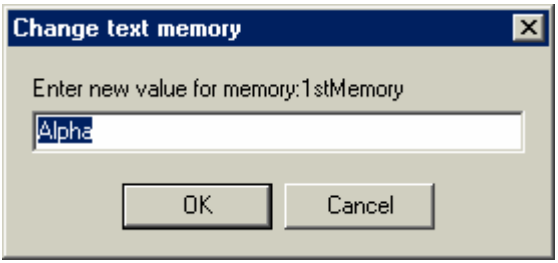
Geben Sie nun den neuen Analogwert ein und bestätigen Sie mit OK.

Simulation:TextMemory

Hier werden alle Textspeicher dargestellt:



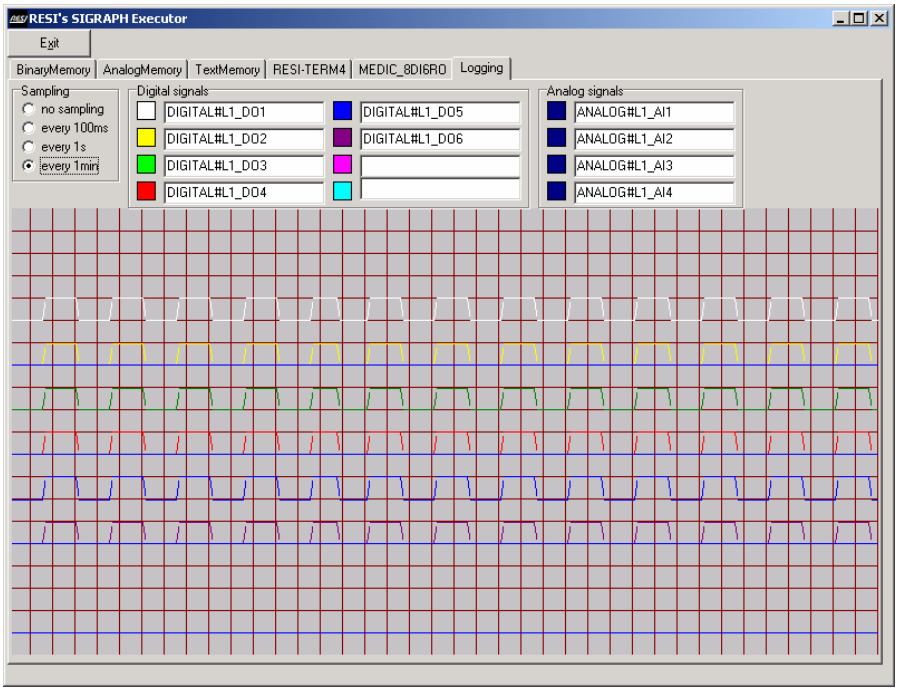
Auch einen Textspeicherinhalt kann man ändern, wenn man auf das Indexfeld doppelklickt. Es erscheint folgende Auswahl:



Geben Sie nun Ihren neuen Text ein und bestätigen Sie mit OK.

Simulation: Logging

Auf dieser Seite kann man die steigenden und fallenden Flanke mit verfolgen:



In diesem Teil des Fensters stellt man die Zeit der Aktualisierung der Flanken ein.

Der Aufruf der gewünschten Digital und Analog Signale:

DIGITAL#L1_DO1

ANALOG#L1_AO1

Simulator beenden

Um den Simulator zu beenden klicken Sie auf Exit.

Kontakt

RESI Informatik & Automation GmbH
Altenmarkt 29, A-8551 Wies, Österreich
Tel: +43-(0) 316-262062-0
Fax: +43-(0)316-262062-66
help@RESI.cc
www.RESI.cc